UK Public Transport Information Profile

A TransXChange profile

for exchanging UK schedule data within the Bus Open Data Digital Service and similar systems

Document Control Log

Date	Modified By	Comment	Version
2020-04-13	SJR	Initial Draft	v0.1
2020-04-14	SJR	Added introduction, adjusted layout & pagination.	v0.2
2020-04-30	SJR	Issue as v1.0 and circulated	v1.0

Table of Contents

List of Figures

List of Tables

1   Introduction

1.1   Scope

The Bus Open Data programme, coming from the Bus Services Act 2017, is placing a requirement on operators to produce TransXChange (TXC) files containing their timetables as open data. While TXC has been around for many years now, and is a mature standard, there are large variety of ways in which data can be constructed within the standard. Some of these are more concise than others, and some inadvertently hide simple data in complex constructs. This makes it hard to interpret data, and users have to be able to interpret the full standard in order to accommodate the different ways that data may be written.

Furthermore, there is a high degree of inheritance of elements permissible within TXC, and this can lead to differences in interpretation, as different systems differ in which elements within a block remain as the default, and which are overridden. This can lead to errors which, when coupled with the complexity of some implementations, are then hard to identify and correct.

Additionally, the continued reliance by the industry on TXC v2.1 has led to a number of errors which are more avoidable if later versions are used. In particular, the coding of trips which run after midnight (operational day vs calendar day principles) is often wrong.

As part of the Bus Open Data (BOD) programme, the Department for Transport (DfT) has sponsored the creation of a TXC profile to support BOD. The aim of this profile is to specify a consistent use of elements and a consistent way of writing TXC that will be used within BOD and which will lead to a higher quality data set and, at the same time, lower the barriers to entry by users new to TXC.

This profile is set out in this document. Because it has a wider applicability than simply the BOD programme, the profile will be referred to as the public transport information profile. It will be abbreviated as the TXC-PTI profile, or simply TXC-PTI within this document.

1.2   Relationship to TransXChange

The TXC-PTI profile will be based on TransXChange version 2.4, as this is the updated version that has most traction and that will be specified in the BOD guidance. In particular, the TXC-PTI profile will be based on the "general" profile and not the "registration" profile. This allows freedom to remove elements that would be mandatory in the registration profile but which are not relevant to TXC-PTI.

It is important to note that the TXC-PTI profile is not a new version of the TXC standard. There is no new "XSD" schema definition file, and anyone creating a TXC v2.4 file will be able to make full use of the standard. All the elements within the schema that are mandatory in the schema will be mandatory in the TXC-PTI profile.

What the profile does, instead, is to specify how that standard is to be used. It specifies additional mandatory elements that are to be used, it specifies which data elements shall *not* be used, and it specifies which are optional. It also details in many instances how the different elements shall be populated, particularly in relation to identifying and versioning different data elements so as to be consistent.

Validation of data against the TXC-PTI profile will take place in two stages:

1) Is the data schema-compliant? This is a straightforward check that checks that the data as submitted is compliant with the basic requirements of the TXC v2.4 schema. This is an in-built function of XML.

2) Is the data TXC-PTI compliant? This is a programmatic check to ensure that the data meets the requirements of this profile, carried out by analysing the data in the TXC document.

Bespoke systems are required to carry out these checks, and are being developed by DfT as part of the BOD programme.

The PTI-TXC profile is an additional set of constraints and clarifications that sit on top of TXC version 2.4. While every care has been taken to ensure compliance with TXC, this document will necessarily need to be read alongside the TransXChange schema guidance document. The most recent published version of this is version 59, which contains information on all versions up to and including TXC v2.5. Elements introduced for v2.5 can be ignored, but users should note that the guidance also contains clarifications and corrections to earlier versions of the schema guidance / standard.

1.3   Abbreviations

BOD	Bus Open Data. The DfT programme emerging from the Bus Services Act 2017.
DfT	The UK Department for Transport.
DVSA	Driver and Vehicle Standards Agency. Responsible for registering bus services.
NOC	National Operator Code. A unique code, usually four characters, that identifies an operator or operating division of an operator and which should be consistent across all systems.
TIL	Traveline Information Limited. A company with overall responsibility for traveline-branded public transport information services in the UK, widely advertised on bus stops and buses. Owners of the NOC database.
TXC	TransXChange. The UK *de facto* standard for timetable interchange.
TXC-PTI	The TXC public transport information profile, set out in this document.
XML	E**X**tensible **M**arkup **L**anguage. The technology used to encode TXC documents.

1.4   Structure of this document

This document is split into sections which form a logical walk through the TXC schema. The sections are not intended to replicate the order that elements occur within a TXC document.

Examples are provided wherever appropriate. As far as possible these are based on real-world examples but should not be taken as representing the operation of actual services or organisations.

Within the document, summaries are placed at key points. These summaries indicate which are the key mandatory items and which are the optional items. They also highlight any key restrictions identified in the text. Summaries are indicated using coloured icons, and these have the following meanings:

	This summarises key mandatory points from the text.		This summarises key optional points from the text

1.5   Credits

This TXC-PTI profile has been sponsored by the DfT, with extensive contributions from the devolved administrations in Scotland and Wales, the UK bus industry, local transport Authorities, and the UK public transport information community. Their input is gratefully acknowledged.

2   General

2.1   General principles

The TXC-PTI profile has been written with a specific set of principles in mind. The aim is to create a simplified profile, which can be used consistently by all systems / operators and where data is clear and unambiguous data. This will lead to data that is more readily understood by downstream open data users, will lead to a higher take up, easier diagnosis, and hence better applications.

The overriding principle is that data should be stated, once, where is best placed. In TXC there is too much scope for writing a mixture of elements, some of which are inherited, some of which are not. There is merit in re-using data where possible, because this keeps file sizes down, but far too often data has been presented where only some of the information in a particular block has been inherited while some has been overridden. This is particularly true of operating profiles, which define the days / dates when the trips actually run.

A secondary, although still important, principle is that data should be stated unambiguously. There are many elements where, for largely historic reasons, ambiguity or shades of meaning have crept in. TXC-PTI aims to remove this ambiguity so that it is clear what the intent of a set of elements or a particular block intend. Again, this is something which is particularly true of operating profiles although it is by no means the only one.

Finally, a third general principle is that ids shall be provided and defined which are consistent from file to file and which provide uniqueness to any given service and line, such that line 8 in one locality is different from line 8 operated by the same operator in a different locality, and that it is clear when a particular line is changing from one version to the next, and which line it is that is changing.

The implementation of these principles will be described more fully in the sections that follow.

2.2   Versions

A common problem with TXC documents up to the present has been the incorrect use of codes to encode versioning information within a file, despite TXC elements generally being provided with a set of attributes which allow this information to be correctly represented. This can lead to, for example, confusion as to whether a file contains a new service or simply a new version of an existing service.

TXC-PTI will therefore adopt some general rules about versioning which shall be followed in those elements which support it. The general principle shall be that, wherever possible, ids and codes used within a series of documents to describe a service shall be consistent from document to document, and that sequences of documents shall be identified and tracked through the documents' version control attributes.

> In all cases where it is supported by the element, version control shall be implemented in element attributes.

Figure 1 - An indicative set of element attributes

- *CreationDateTime* shall be populated with the date and time at which the data was first created and shall not be changed in any future version of the same element. Where different versions of data appear in different files, this shall require the generating system to keep track of the version history.

- *ModificationDateTime* shall be populated with the date and time of the current data. It shall be omitted for the original (version 0) issue of the data, but shall be present in all subsequent versions of the same data.

- *Modification* is a status flag and shall be set to either "New" for the first issue of the data, or to "Revise" for subsequent versions of the same data. Data providers may, if they wish, continue to publish data once it has expired, and may do so using appropriate alternate *Modification* enumerators. However, this is not expected to be a normal use case for TXC-PTI.

- *RevisionNumber* is a sequence number that shall be 0 for the first issue of the data, and shall increment by 1 for each subsequent version of the data. It is important to note that this revision number is used to track TXC-PTI data from version to version and is independent of any other versioning system. For example, an operator may wish to update operational

dates to accommodate school term changes. This would require a change of revision number in the TXC-PTI data, but would be independent of, and not reflect the current value of, the revision number recognised by DVSA in respect of the service registration.

- *Status* is of limited value since data elements (e.g. **Service/OperatingProfile**) will describe the extent to which data is active. There is therefore no requirement to provide this attribute, and for any conflicts between *Status* and data, the *Status* flag shall be ignored.

2.3   Accessibility information

Early version of TransXChange schemas had no facility for adding information about vehicles, particularly with respect to wheelchair accessibility and on-board facilities for passengers with visual and/or other impairments. TXC version 2.4 was the first version to include this information, and this was then later extended by TXC version 2.5. While TXC-PTI is based on v2.4 rather than the later version, it can nevertheless make use of the facilities that are in v2.4 to describe the type of vehicle and some of the accessibility features of the vehicle.

This is done through the **VehicleType** element, where:

- **VehicleTypeCode** and **Description** are operator-defined elements that describe the vehicle. For example, "LFDD" and "Low Floor Double Decker". These are textual elements and are only used for visual reference, rather than any direct use in e.g. journey planning, and so can be omitted.

- **VehicleEquipment** is an element which provides details about equipment available for passenger information (**PassengerInfoEquipment**), information regarding the accessibility of the vehicle itself, such as whether low floor or fitted with a ramp (**AccessVehicleEquipment**), and further information on wheelchair spaces, including how many and whether they need pre-booking and how (**WheelchairVehicleEquipment**).

- **WheelcharAccessible** is a boolean true/false flag that states in simple terms whether the vehicle is accessible to passengers in a wheelchair. This flag should be stated explicitly in order to avoid doubt over the meaning of its omission.
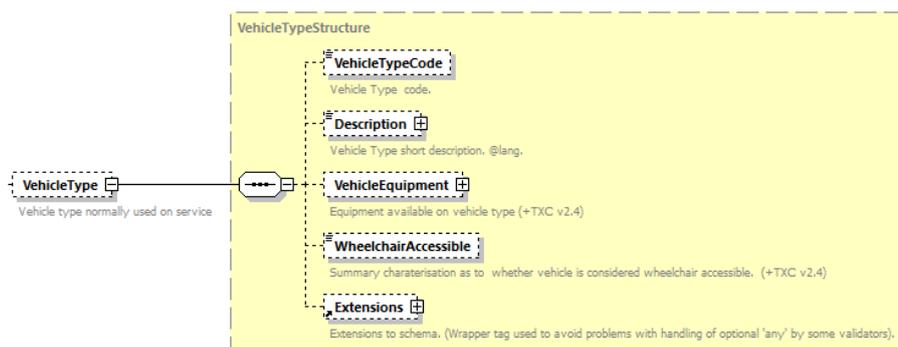
Figure 2 - The VehicleType element

The vehicle equipment block is described more fully below.

There is no specific requirement in TXC-PTI to specify this information. However, the Bus Open Data programme will require operators to provide much of this equipment on vehicles and it is recommended that operators consider providing this information now as part of the TXC-PTI document.

Note that, irrespective of the data contained in a TXC-PTI document, there is no obligation on the operator beyond those set out in law to provide any of the equipment listed. Operational issues will determine whether any particular trip will operate as stated. The best that the TXC-PTI document can do is indicate what trips are likely to be available to people with reduced mobility or other impairments.

> It is recommended that a TXC-PTI document should provide, as far as possible, the information in the **VehicleType** element to facilitate journeys by people with reduced mobility or other impairments.

2.3.1   Passenger information equipment
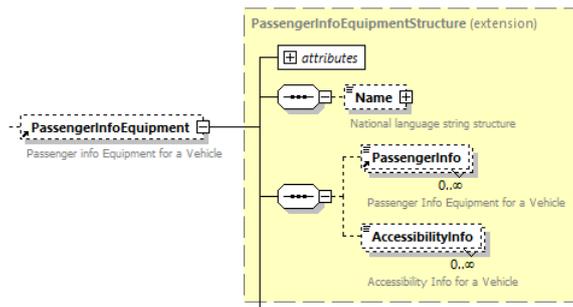
Figure 3 - The PassengerInfoEquipment element

Passenger information equipment describes, through a set of elements which are classified as either passenger or accessibility information, the facilities that the vehicle has for presenting passenger information. The allowable values are shown in Table 1 below.

Table 1 - Permitted values for PassengerInfoEquipment

Group	Permitted Values	Used in TXC-PTI
PassengerInfo	nextStopIndicator	Where appropriate
	stopAnnouncements	Where appropriate
	passengerInfoFacility	Not used
	other	Not used
AccessibilityInfo	audioInformation	Where appropriate
	audioForHearingImpaired	Where appropriate
	visualDisplays	Where appropriate
	displaysForVisuallyImpaired	Where appropriate
	tactilePlatformEdges	Not used
	tactileGuidingStrips	Not used
	largePrintTimetables	Not used
	other	Not used

For example, a vehicle which has a "next stop" indicator and audio information might be described as follows:

```
<PassengerInfoEquipment>
   <PassengerInfo>nextStopIndicator</PassengerInfo>
   <PassengerInfo>stopAnnouncements</PassengerInfo>
   <AccessibilityInfo>audioInformation</AccessibilityInfo>
   <AccessibilityInfo>audioForHearingImpaired</AccessibilityInfo>
</PassengerInfoEquipment>
```

2.3.2   Vehicle access equipment
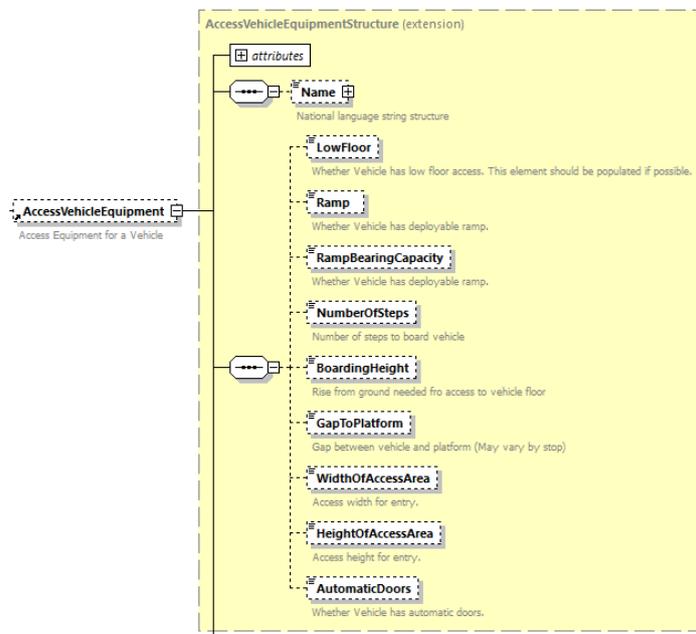
Figure 4 - The AccessVehicleEquipment element

The **AccessVehicleEquipment** element describes how the vehicle may be accessed. A number of elements, such as the gap to the platform are not especially useful in a bus context, while others such as low floor and ramp flags are important. Their usage for TXC-PTI is shown in Table 2 below.

Note that for both LowFloor and Ramp, it should be stated explicitly whether these features are present or not. This avoids any doubt which may arise from their omission.

Table 2 – Permitted elements for AccessVehicleEquipment

Element Name	Data Type	Used in TXC-PTI
LowFloor	*boolean*	Yes. State explicitly whether true or false.
Ramp	*boolean*	Yes. State explicitly whether true or false.
RampBearingCapacity	*Weight*	Not used
NumberOfSteps	*nonNegativeInteger*	Where appropriate
BoardingHeight	*Length*	Not used
GapToPlatform	*Length*	Not used
WidthOfAccessArea	*Length*	Not used
HeightOfAccessArea	*Length*	Not used
AutomaticDoors	*boolean*	Not used

For example, a vehicle which has a low floor, but which does not have a deployable ramp for wheelchair access, would be coded as:

```
<AccessVehicleEquipment>
    <LowFloor>true</LowFloor>
    <Ramp>false</Ramp>
</AccessVehicleEquipment>
```

2.3.3   Wheelchair-specific information
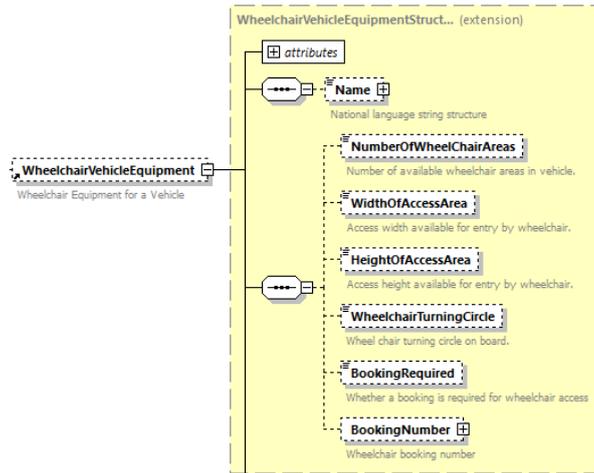
Figure 5 - The WheelchairVehicleEquipment element

The **WheelchairVehicleEquipment** element provides information on the number and size of wheelchair spaces in the vehicle, together with information on whether or not advance booking is required, and the number to call if it is. Their usage for TXC-PTI is shown in Table 3 below.

Table 3 - Permitted elements for WheelchairVehicleEquipment

Element Name	Data Type	Used in TXC-PTI
NumberOfWheelChairAreas	*nonNegativeInteger*	Where appropriate
WidthOfAccessArea	*Length*	Not used
HeightOfAccessArea	*Length*	Not used
WheelchairTurningCircle	*Length*	Not used
BookingRequired	*boolean*	Where appropriate
BookingNumber	*TelephoneContactStructure*	Yes, if BookingRequired is true

For example, a vehicle with 2 wheelchair spaces which require booking may be coded as follows:

```
<WheelchairVehicleEquipment>
    <NumberOfWheelChairAreas>2</NumberOfWheelChairAreas>
    <BookingRequired>true</BookingRequired>
    <BookingNumber>
        <TelNationalNumber>01234 567890</TelNationalNumber>
    </BookingNumber>
</WheelchairVehicleEquipment>
```

2.4   Notes

Notes are permitted in various places in TXC-PTI and follow the same structure as in a standard TXC document.
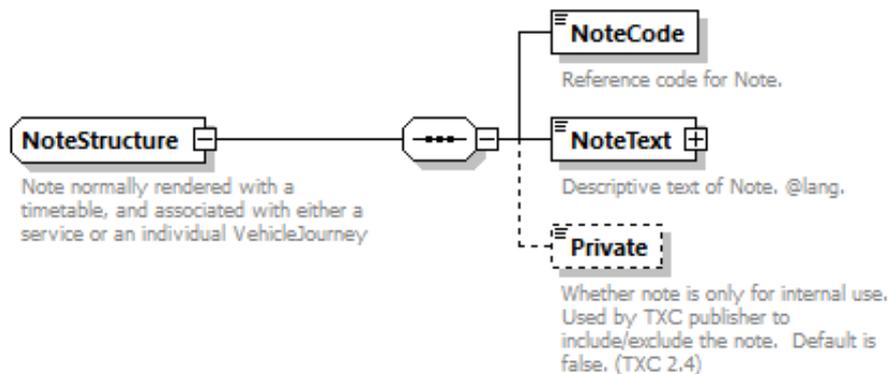
Figure 6 - The Note structure

There are no specific requirements for notes that arise within TXC-PTI, except that:

a)   notes should not be flagged as private; and

b)   notes should not encapsulate anything in a text format that is best described through data. This is because consuming systems cannot process textual data without a high degree of effort.

An example of an inappropriate note might include, for example, "Starts 3 minutes earlier at Cefn Coch Inn", as this should be encoded with suitable stop and timing records.

> For TXC-PTI, **Notes** are permitted as appropriate, but shall not encapsulate information that is best described through data.

3   Serviced organisations

3.1   Introduction

TXC has a mechanism for holding the dates when organisations, such as schools, are open and closed. This is the **ServicedOrganisation**. This gives one place where a common set of dates can be defined, and these can then be applied to individual vehicle journeys by a straightforward reference to the serviced organisation code. A **ServicedOrganisation** also contains a code by which the organisation can be referred, and a description.

This makes a serviced organisation very useful in terms of individual trips. Rather than coding them as operating date records in a **VehicleJourney**, reference to the **ServicedOrganisation** allows them to be given context when presenting them to users either as part of a journey plan or via a traditional timetable, for example "operates on Dame Alice Harpur school days only".

For this reason, TXC-PTI encourages the use of serviced organisations.

However, operators have expressed strong reservations about their ability to manage dates for each organisation as, while there may well be a core of commonality, many different schools and colleges will have different starts and ends to terms and will have different inset (non-teaching) days. Several schools and colleges may be featured on the same trip, and this additionally gives operators

difficulties in knowing how to capture the description of when a particular trip runs; it may well be the case that, while one organisation is still open, another has already closed.

Nevertheless, an operator must know whether they are operating a particular trip or not, even if they cannot describe it through a **ServiceOrganisation**. For TXC-PTI, then, a **ServicedOrganisation** is encouraged for the context that it can add, but is not mandatory. However, all trips shall, either through serviced organisations or through dated records, carry the correct dates so that it is unambiguous whether a trip runs or not on any particular day.

> In TXC-PTI, the use of **ServicedOrganisations** is strongly recommended but is not mandatory.

> In TXC-PTI, all trips shall be coded to state unambiguously when they operate. If **ServicedOrganisations** are not used, then appropriate date records shall be used.

3.2   Defining a serviced organisation

A **ServicedOrganisation** has, at first glance, a significant number of elements that can be populated (see Figure 7). TXC-PTI will not use most of these, however, as they are of little use to the public; they carry such information as contact details and classification of a single school, or refer to NaPTAN or NPTG codes that are not public-facing. Therefore, the elements listed below shall be the only fields populated.

Permitted elements and their values in TXC-PTI are:

- **OrganisationCode**, which is mandated by the schema. There is a recommendation in the schema that this should be the reference number of the school, as allocated by the administrations in the devolved governments, however this is not required by TXC-PTI. Firstly, a textual code can be more meaningful (e.g. COWA = College of West Anglia), and secondly, as has already been noted, a set of dates could cover multiple organisations.

- **Name** shall be provided, and shall be a meaningful name for the organisation, or group of organisations, being referenced. For example, "College of West Anglia" or "Staffordshire schools".

- **ServicedOrganisationClassification** may be provided if additional context is required. This is an enumeration that describes the type of establishment, e.g. "school", "factory", etc. However, not all systems will recognise or use this.

- **WorkingDays**, which provide a series of date ranges when the organisation is open. This date range:

  o   shall have an inclusive **StartDate**;

  o   shall have an inclusive **EndDate**;

  o   may have a **Description** to give some context to the date range, although a description is not mandatory in TXC-PTI; and

  o   may describe a date range as **Provisional**, although such date ranges must be replaced with actual date ranges prior to the operation of the service.

Note that **Holidays** shall not be provided. Providing both working dates and holiday dates might seem like a benefit, but there have been instances of overlapping date ranges, and strange

encodings e.g. where a service is said to not run in the holidays of an organisation. This can lead to confusion and misinterpretation of the data.
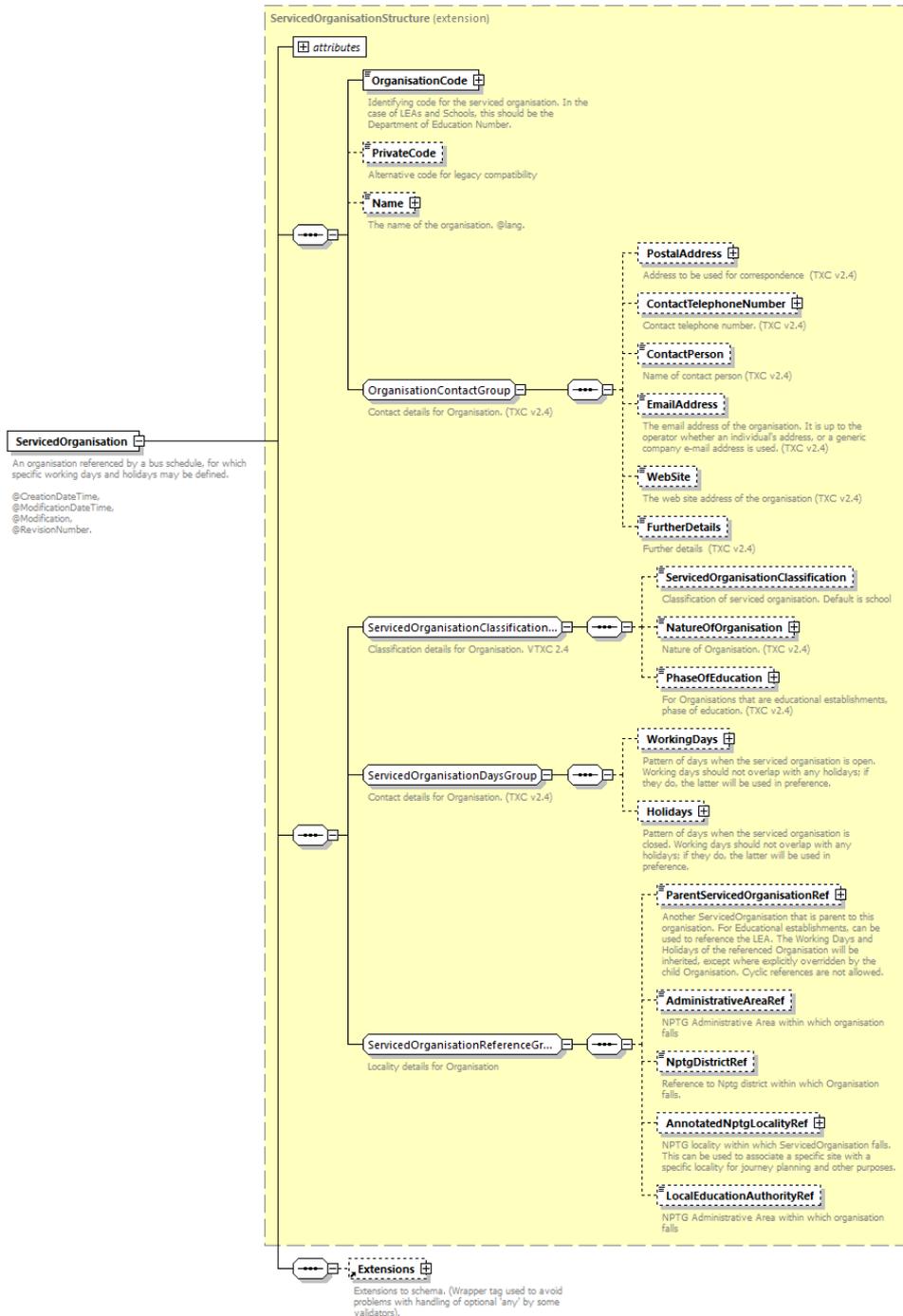
Figure 7 - The ServicedOrganisation element

In TXC-PTI, therefore, only **WorkingDays** shall be defined, and all other days that are not **WorkingDays** shall be assumed to be holidays. This means that, in a **ServicedOrganisationDayType** within an operating profile, trips which only operate on working days will be encoded as **DaysOfOperation/WorkingDays**, while trips which only operate during the holidays will be encoded as **DaysOfNonOperation/WorkingDays**. It is recommended, however, that any presentation to the

user avoids negative descriptions such as "this trip does not operate on the working days of X" but presents it instead with a positive description such as "this trip operates during holidays of X."

See section 9.3 for more details of the **OperatingProfile** element, and in particular the relationship between the working dates and the regular days of operation in section 9.3.6.

In TXC-PTI, a **ServicedOrganisation** shall be given a code, a meaningful name, and working dates when the organisation is open. Holiday dates shall not be provided; all non-working dates shall be assumed to be holidays.

3.3 Example of a serviced organisation coding in TXC-PTI

The following is an example of how a **ServicedOrganisation** may be coded and used within a TXC-PTI document. For further details about how the data is encoded in an **OperatingProfile**, please see section 9.3.

```xml
<ServicedOrganisation>
    <OrganisationCode>BurCol</OrganisationCode>
    <Name>Burnley College</Name>
    <WorkingDays>
        <DateRange>
            <StartDate>2019-04-22</StartDate>
            <EndDate>2019-05-24</EndDate>
        </DateRange>
        <DateRange>
            <StartDate>2019-06-03</StartDate>
            <EndDate>2019-06-28</EndDate>
        </DateRange>
    </WorkingDays>
</ServicedOrganisation>
...
<VehicleJourneys>
    <VehicleJourney SequenceNumber="1">
        <PrivateCode>120MFBP03:O:1</PrivateCode>
        <OperatingProfile>
            <RegularDayType>
                <DaysOfWeek>
                    <Monday/>
                    <Tuesday/>
                    <Wednesday/>
                    <Thursday/>
                    <Friday/>
                </DaysOfWeek>
            </RegularDayType>
            <ServicedOrganisationDayType>
                <DaysOfOperation>
                    <WorkingDays>
                        <ServicedOrganisationRef>BurCol</ServicedOrganisationRef>
                    </WorkingDays>
                </DaysOfOperation>
            </ServicedOrganisationDayType>
        </OperatingProfile>
        ...
    </VehicleJourney>
    ...
</VehicleJourneys>
```

4   Operators, Garages and Registrations

4.1   Introduction

In a TransXChange document, every trip that operates is operated by an **Operator**. Given that TXC is most often used for bus and coach data, an **Operator** is usually a registered bus or coach operator in possession of a valid, current, DVSA-issued "O-licence". However, it is also possible that an **Operator** is a not-for-profit community transport association with a "Section 22" permit, or an operator of another mode. The schema is agnostic as far as the type of operator is concerned.

4.2   The Operators group

The **Operators** group in TXC is a grouping element that can hold one or multiple operator records, classified into different operating categories: **Operator** or **LicensedOperator**. There is no difference in the content of these two categories; it is simply that **LicensedOperator** contains some additional mandatory elements above what is required for an **Operator**. However, a **LicensedOperator** includes some mandatory information that is better sourced from other sources, and hence only **Operator** shall be used.

The Bus Open Data requirements that will be supported by the TXC-PTI profile are for each operator to be responsible for their own data. Therefore, an **Operators** group shall only contain a single instance of **Operator**.

For TXC-PTI, an **Operators** group shall contain only a single **Operator**.
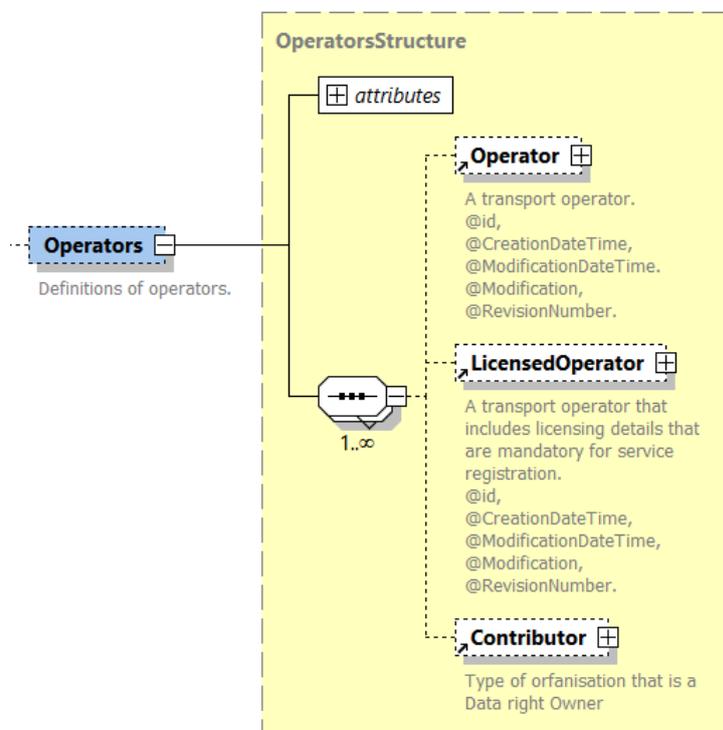
Figure 8 - The Operators structure

4.3   The Operator element

The **Operator** element contains a large amount of information about the operator, most of which is best obtained from other sources by means of referenced identities. There is no need to replicate it within the TXC, and this also avoids the risk that conflicting information is provided; there needs to be a single source of truth. The **Operator** element therefore only needs to contain references which link to those external sources, and such information as is either mandatory in the schema or which cannot necessarily be inferred from external sources.
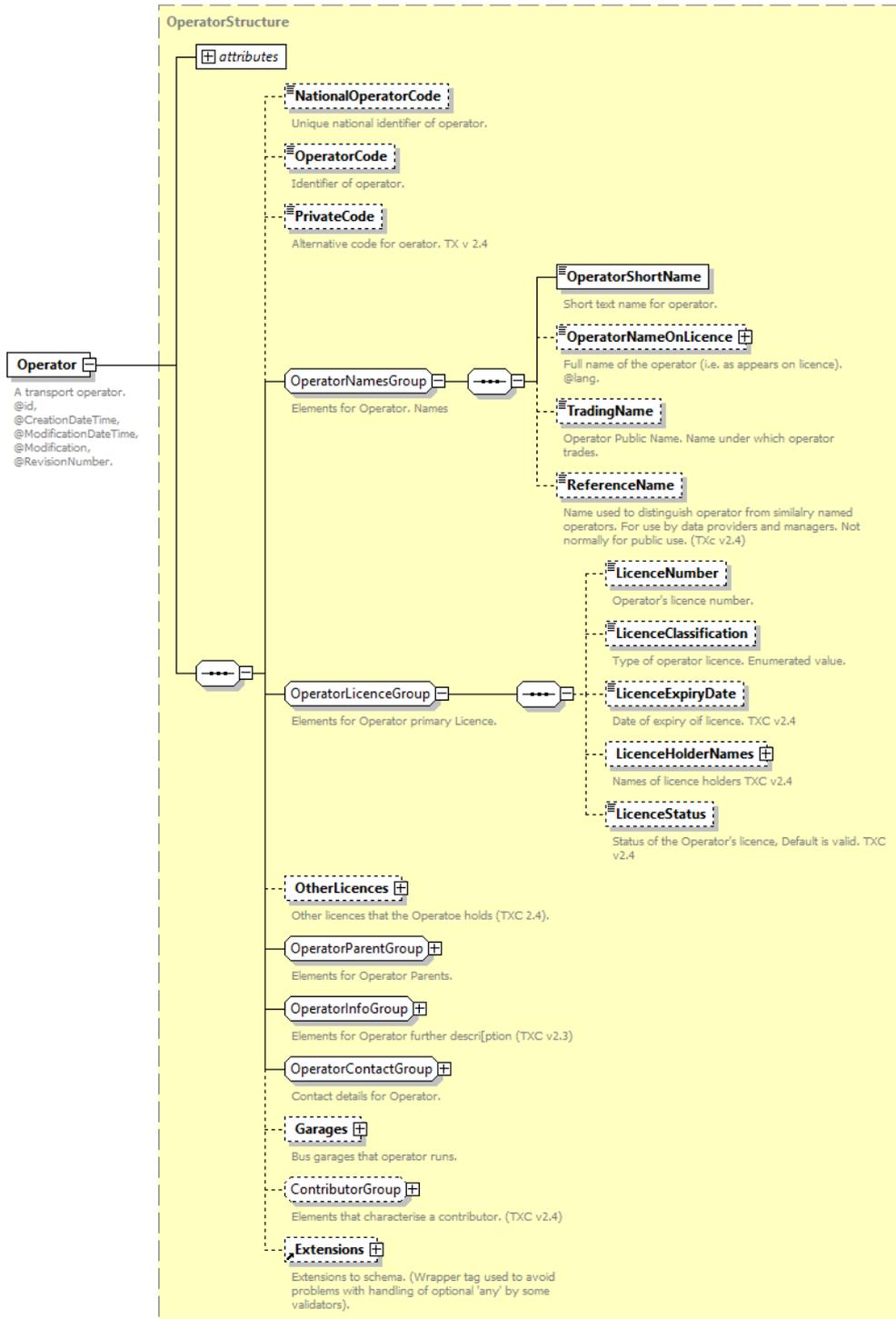
Figure 9 - The Operator element

Accordingly, the following shall be adopted for TXC-PTI:

- Each **Operator** shall have an *id* attribute. This is an internal reference only and is only used to reference other elements within the TXC document

- **NationalOperatorCode** is mandatory. This code shall refer back to the appropriate code held within the National Operator Code database (NOC) maintained and operated by Traveline Information Limited (TIL) or its successor organisations. The publicly accessible NOC database shall hold all the appropriate details for the operator including, but not limited to, contact details, websites, public-facing operator identities, etc. as appropriate. The NOC code shall be used as the key for accessing this information, without it having to be duplicated within the TXC document except as outlined below.

- **OperatorCode** is not required, although it may be populated for compatibility with other systems

- **OperatorShortName** is mandated by the schema. Where possible this should match the Operator Public Name field in NOC, although adjustments may have to made in order to accommodate the maximum 24-character length of this field in TXC. Responsibility for how this is done shall lie with the operator.

- **TradingName** shall be provided, where this is different from the name given in NOC.

- **LicenceNumber** shall be provided where applicable. Where an operator has more than one O-licence, the licence number provided shall be the licence number relevant to the service being described in the TXC document. Details about the licence shall be obtained via reference to DVSA, and not to other sources, although **OperatorNameOnLicence** may be provided to aid human readability of the TXC document.

- **Garage** entries in the **Garages** group may be provided, but only inasmuch as they are required for use in e.g. dead runs in **VehicleJourneys** and for use in real time systems. Otherwise, this information shall be left empty.

- No other elements shall be populated for **Operator**. This includes, specifically, any contact information contained within the **OperatorContactGroup** as this information shall be taken from NOC referenced by the **NationalOperatorCode**.

4.4   Registrations

In EBSR documents, a TXC **Registrations** block is included to replicate the information that would be contained on the paper registration. However, for TXC-PTI which is public-facing information, this information is superfluous and adds nothing to the quality of the open data. For this reason, the **Registrations** block shall not be included in a TXC-PTI document.

There is merit in referencing the DVSA registration reference, however, for traceability and completeness purposes, but this shall be encoded as part of the **ServiceCode** within the **Services** element (see 5.3.2 below).

The **Registrations** group shall not be included within a TXC-PTI compliant document.

5   Services and Lines

5.1   Introduction

In the UK, the label "service" is often used in common terminology to denote a particular route, for example service 1. However, some people will take this to mean the 1 and all its variants (which may, perhaps, be called 1A, 1B, X1, etc.) while some will take it to mean only the 1, and that 1A, etc. are different services. Both can be correct, depending on local usage and custom, but in data it is important to make the distinction.

In TXC, this distinction is made using the terms "service" and "line", where:

- A LINE is a collection of vehicle journeys (trips) that share some high degree of commonality of route and timings, and which are all known by the same LINE NAME. Examples are "1", "1B", "100", "calverton connection", etc. (there is no requirement for a LINE NAME to include a number)

- A SERVICE is a collection of LINEs that together make up a coherent set of vehicle journeys usually running over geographically similar routes although this isn't a requirement. For example, consider a service "1/1A", where line "1" operates "A to B direct" and line "1A" operates "A to B via C", and together they make up the complete service between A and B.

Note that this description and example should be taken as illustrating the terms and the concept only – there is no requirement in TXC-PTI for how an operator should combine, or not, services and lines together in a TXC document.

5.2   The Services group

The **Services** group in TXC is a group that holds one or more **Service** elements in a *ServicesStructure*. It has been common practice in some operators' data to submit several services within the same TXC document. However, there is often little consistency in how these are grouped and, often, multiple services contain single lines where the intent would be better represented by a single service with multiple lines. Additionally, including too much information in a single TXC document can make it harder to fault-find a single service due to the inclusion of data not related to that service. In TXC-PTI, therefore, the permitted number of services in a single TXC document shall be limited.

 In TXC-PTI, the **Services** group shall only contain a single **Service** element.

5.3   The Service element

5.3.1   Introduction

The **Service** element in TXC describes the service, the lines that it comprises, and provides other information about the default operation of the service. It has a number of different elements that are schema-mandated, and these will of necessity be required. However, TXC-PTI will provide rules as to the ways that a number of these shall be populated.

As stated earlier, there shall only be one **Service** element in TXC-PTI file. This applies even to different versions of the same service, because a duplication would require different keys to be contained within the data, which would then break the requirement for the keys to be consistent from version to version.

A **Service** shall be versioned, as described in 0 above.

5.3.2   Identifying the service
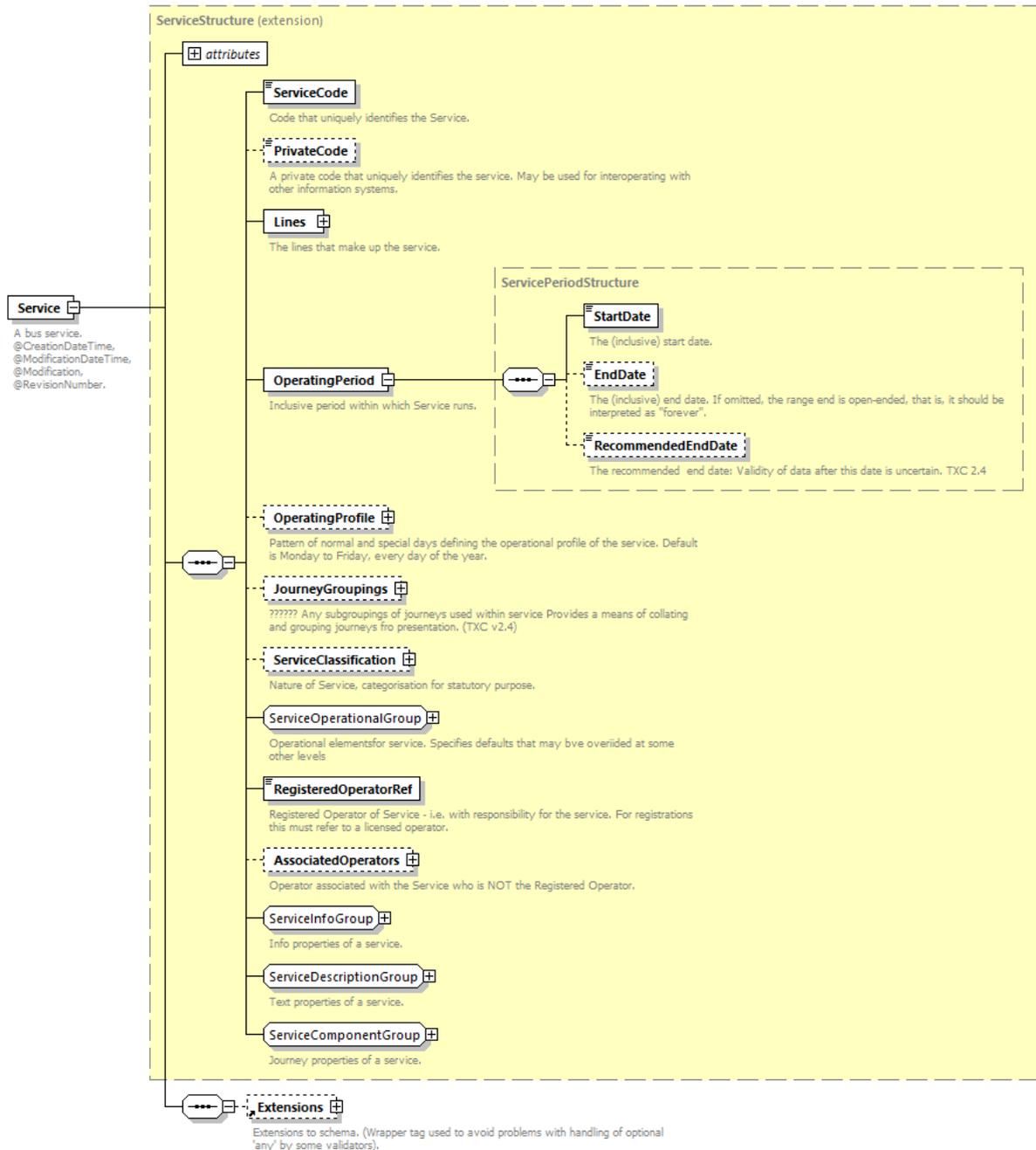
Figure 10 - The Service element

The **ServiceCode** element uniquely identifies the service. As far as possible, for TXC-PTI this shall be the registration reference as given when the service is registered with DVSA. For a DVSA registration this takes the form of the "O-licence" number, a forward slash, and a sequential number, e.g. PF0000459/134 is the registration reference for Stagecoach service 3 in Bedford. Using the registration reference for the **ServiceCode** in this way facilitates traceability and permits other operations such as completeness checks to be carried out.

While a forward slash is not a reserved character in XML, to avoid confusion and to aid readability the **ServiceCode** in TXC-PTI shall replace the forward slash in the registration reference with a colon.

```
<Service>
   <ServiceCode>PF0000459:134</ServiceCode>
   ...
</Service>
```

While Bus Open Data is directed initially towards operators of registered bus services, the TXC-PTI profile is intended to fulfil a wider use. Data for unregistered services therefore needs to ensure that **ServiceCode** is a) unique; and b) consistent from version to version. In TXC-PTI, uniqueness for unregistered services shall be ensured by formatting the **ServiceCode** as follows:

- Start with the letter code "UZ" to denote an unregistered service. The second letter is a placeholder in case a future version of this profile requires to split out unregistered codes e.g. by region in a similar way to DVSA's "PF", "PH", etc. O-licence codes.

- Follow with three zeros and then the 4-letter NOC code of the operator (or four zeroes if the NOC code is three letters).

   These two elements combined provide a format which is the same length and styling as the O-licence for a registered service.

- Follow that with an operator-generated reference number that is unique for that service, and which shall persist from version to version of the same service. Separate this with a colon, as is the case for the registered services.

For example, West Norfolk Community Transport may choose to encode their "Go To Town" service 32 like this:

```
<Service>
   <ServiceCode>UZ000WNCT:GTT32</ServiceCode>
   ...
</Service>
```

To reiterate, the last part of the code is entirely down to the operator. There is no requirement to include the line number, as shown in the example, as the line number or name will be included in the *id* of the Line - see section 5.5 below.

> In TXC-PTI, the **ServiceCode** shall be the DVSA-provided registration reference of the service, with the forward slash replaced with a colon.
>
> Where a service is unregistered, operators shall use a similar format with the O-licence number being replaced with a pseudo-reference made from "UZ" plus the operator's NOC and a unique reference as described above.

5.3.3   Operating dates and patterns

In TXC-PTI, the overall dates of operation of the service shall be defined, as with the full TXC schema, by the mandatory **OperatingPeriod** element. As a minimum, an **OperatingPeriod** requires a **StartDate** and, if the service is terminating in the near future (because there is an impending change, or because the service being described only operates for a necessarily limited period e.g. roadworks shuttle), an **EndDate**. If the service is expected to run "until further notice" then the **EndDate** shall be omitted.

In TXC-PTI, the use of an **EndDate** in the distant future (which may, or may not, change from revision to revision) shall not be permitted. An **EndDate** shall only describe the actual end date of a service.

Between these (inclusive) start and end dates, the default operation is described by the **OperatingProfile**. In standard TXC documents, this is predominantly used in one of two ways:

a) To describe the overall operation of the service, e.g. "Mondays to Sundays" for a seven-day service, even though different timetables may operate Mondays to Fridays, Saturdays, and Sundays; or

b) To describe the majority of trips e.g. "Mondays to Fridays" even though there are also trips which run on Saturdays and on Sundays.

A principle behind TXC-PTI is that data which is declared as a default and then over-ridden in a more specialised element should be avoided wherever possible, as this leads to confusion. On that basis, it is preferable that the **Service/OperatingProfile** element should be omitted. However, this does mean that the size of files could expand dramatically when there is no need to because all **OperatingProfile** information has to be encoded into individual **VehicleJourney** elements.

For TXC-PTI, therefore, this principle can be relaxed for **OperatingProfile**, so long as the operating profile completely defines the operations of the largest number of trips in full (e.g. it defines Mondays to Fridays excluding Bank Holidays). Individual **VehicleJourney** elements then have either:

a) no **OperatingProfile**, because they follow the default pattern; or

b) a complete set of **OperatingProfile** elements which completely replace the default pattern, and which describe in full when that **VehicleJourney** runs.

Where used in a **Service**, the **OperatingProfile** shall follow the rules described in section 9.3.

In TXC-PTI:

- the **OperatingPeriod** shall only include an **EndDate** if the service has an actual end date.

- The **OperatingProfile** may only describe, in full, the default days of operation and non-operation of the service, which shall be applicable to the majority of the trips. Otherwise, it shall be omitted.

- **OperatingProfile** may not describe the overall operation of the service.

5.3.4   Service operated by multiple operators

The definition of a service, as stated in the introduction to this section, is that it is a collection of lines that together make up a coherent set of vehicle journeys. The example used in the introduction was of line 1 and line 1A, highlighting one use case of two lines operated by the same operator. However, there is a second class of service, which is one that is operated by multiple operators.

For example, many services are operated as commercial services by operators Monday to Saturday, from early morning through to early evening, with local authorities then contracting a second operator to run evening trips, or trips on Sundays. Another example is a quality partnership, such as is found in Oxford, with two operators alternating trips on the same line to provide a high frequency service to passengers. In both these cases, the two operators' individual services together make up the overall service that the passenger sees.

The Bus Open Data programme requires operators to be responsible for their own data, and the TXC-PTI profile confirms this by only allowing one operator per TXC-PTI document, with reference back to the single-operator service registration and O-licence. It is therefore not possible in TXC-PTI

to capture the entirety of a service within a single document when it is operated by different operators.

In order to ensure, as far as possible, that the linkage between the different operators' services is maintained, use should be made within TXC-PTI of the facilities to identify the alternate operators and the alternate services. This is also a suitable mechanism for a single operator to use where there is a split registration with different sections in separate documents based on registrations, and where the operator wishes to indicate that they are linked.
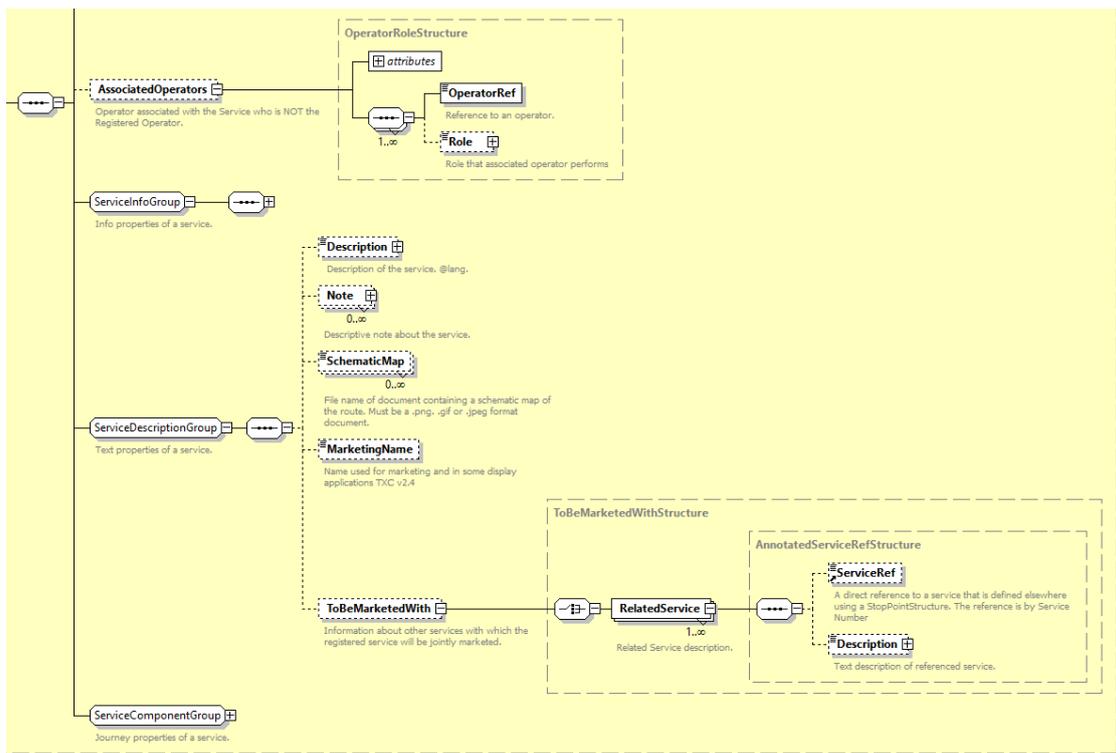
Figure 11 - Excerpt from Service element showing association elements

Within **Service**, this linkage can be shown using the **AssociatedOperators** element, which identifies the operator(s) that are linked, and **ToBeMarketedWith**, which identifies the service(s) that should be linked. Together, these give a complete set of information to enable end users to link services, although there is no direct way to link one of a number of associated operators with a specific linked service.

- **AssociatedOperators** contains a list of **OperatorRef** elements. **OperatorRef** is a text entry that, in TXC-PTI, shall be the national operator code of the associated operator.

- **ToBeMarketedWith** contains a list of **RelatedService** elements. A **RelatedService** element itself can, in TXC, contain a **ServiceRef** and/or a **Description**. However, unlike the **OperatorRef** in **AssociatedOperators**, a **ServiceRef** has to refer to a service defined within the same TXC document. For TXC-PTI, therefore, a **RelatedService** shall only contain a Description element which shall be populated with the external service reference (**ServiceCode**) of the linked service.

For example, the code snippet below shows a service which is related to one operated by the operator with a national operator code of "PQRS", and specifically refers to a service with service code PH0001045:22. It is the responsibility of the downstream user to identify this service and to fetch the data for it.

```xml
<AssociatedOperators>
   <OperatorRef>PQRS</OperatorRef>
</AssociatedOperators>
...
<ToBeMarketedWith>
   <RelatedService>
      <Description>PH0001045:22</Description>
   </RelatedService>
</ToBeMarketedWith>
```

In TXC-PTI, services which are operated by multiple operators may be linked using **AssociatedOperators** and **ToBeMarketedWith**. The **OperatorRef** shall be the NOC of the linked operator, and the **Description** for the **RelatedService** shall contain the **ServiceCode** of the linked service.

This mechanism may also be used to link different documents (services) from the same operator where there is a split registration.

5.3.5   Standard services
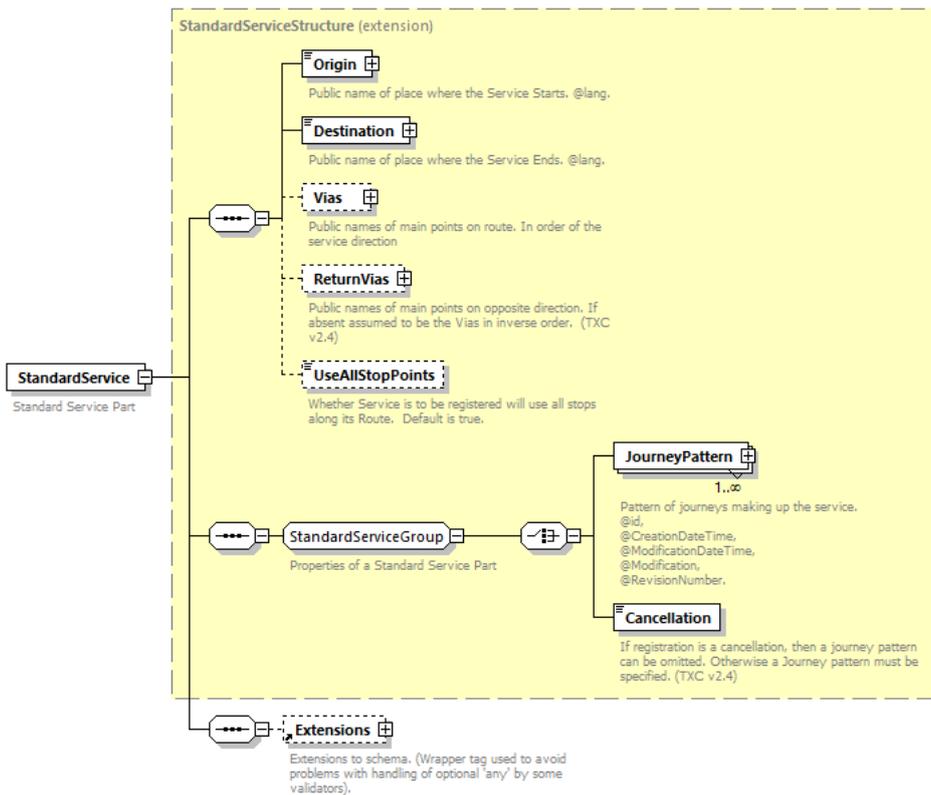
Figure 12 -The StandardService element

Within the Service element, TXC can describe the default operating pattern of the service, either as a standard service or as a flexible service. A standard service is one which follows a set of stops, in sequence, to a strict timetable, while a flexible service is one which may or may not follow a set of stops in sequence, but operates flexibly as and when it is booked. It may therefore not run at all and may only run to some of the stops that are declared.

TXC-PTI is only concerned at present with standard services, however flexible services are discussed briefly in section 10 for future extensions of the profile.

A **StandardService** element is theoretically optional. However, there needs to be at least one **JourneyPattern** defined because at least one standard (non-flexible) **VehicleJourney** needs to reference a journey pattern. For a non-flexible service there shall therefore be a **StandardService** element in a TXC-PTI document, which:

- shall include **Origin** and **Destination**, as these are mandated by the schema;

- may optionally include **Vias** and **ReturnVias**, however these are better completed for the individual lines and hence can largely be ignored for TXC-PTI;

- shall not include the **UseAllStopPoints** flag, as this is not public facing information and all the stopping points that the service uses should be encoded into the individual vehicle journeys via references to journey patterns.

- shall include one or more **JourneyPattern** elements sufficient to describe at least one pattern in each direction. For details of how a **JourneyPattern** is to be structured, refer to section 8.2 below8 below.

> In TXC-PTI, a non-flexible service shall have a **StandardService** which shall consist of at least one **JourneyPattern**.

5.3.6   Interchanges

5.3.6.1   Introduction

Interchanges in TXC can be problematic, because all of the trips that are required for interchange need to be in the same TXC document. This means that, in many cases and particularly for split registrations, it is not easy to describe the interchanges. Other methods of communicating this data have been therefore been developed, although these show some supplier variation and are not part of a standardised TXC exchange.

However, this does not mean that TXC is incapable of describing interchanges, or that TXC-PTI documents should not try to describe interchanges when all of the required information is already in the document. This is particularly true for circular and "pan-handle" services where the outbound trips merge seamlessly into the inbound trips (and/or *vice versa*) and where passengers can remain in the vehicle. Here, the relevant trips would all be in the same document, since this is a single line.

A secondary class of interchange of this type is where a line name changes to another line name using the same vehicle, and again passengers are allowed to remain on the vehicle, or where there is a guaranteed connection between two lines albeit that passengers have to change (e.g. the vehicles "kiss and turn"). These secondary usages are more limited in scope, because the two lines would require to be in the same document and that cannot be guaranteed.

However, wherever interchanges can be defined, it is recommended that a TXC-PTI document should provide this information using TXC's **JourneyPatternInterchange** element.

> It is recommended that, in TXC-PTI, interchanges which can be made on the same service should be included using **JourneyPatternInterchange** elements as described below.

The **JourneyPatternInterchange** element is the way that interchanges are described at a service level in a TXC file. This describes interchanges that can always be made, regardless of the trip

operating. Where possible connections exist that vary from trip to trip (for example, only some journeys are through journeys) then this should be described by a **VehicleJourneyInterchange** instead (see section 9.6).
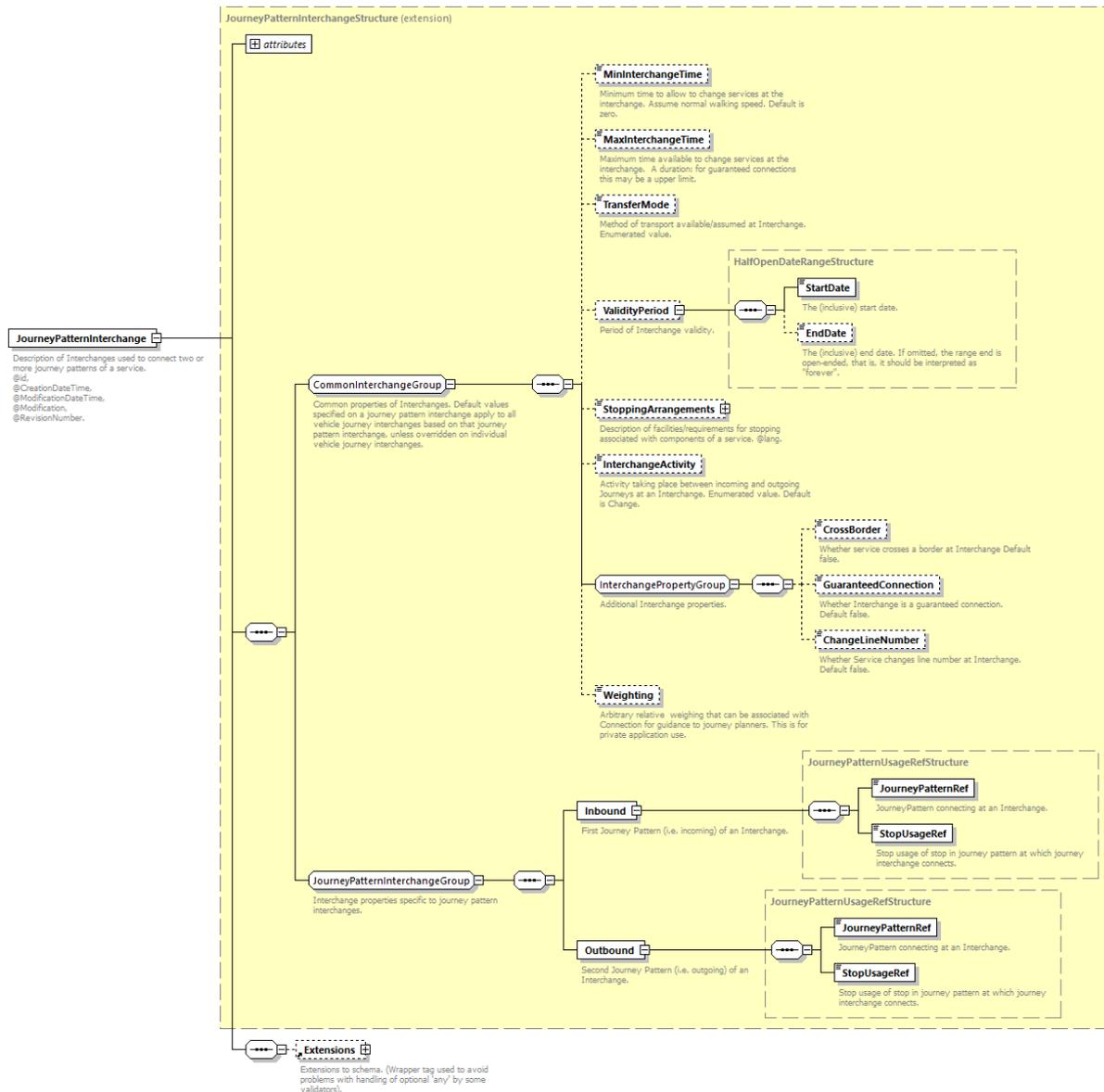
Figure 13 - The JourneyPatternInterchange element

A **JourneyPatternInterchange** is made up of two groups of elements. The first of these is the common interchange group, which is a set of elements that are common to both journey pattern interchanges and vehicle journey interchanges, and the second of these is the journey pattern interchange group, which is a set of elements that are particular to a journey pattern interchange.

Where a TXC-PTI document chooses to have default journey patterns with zero-minute link durations and actual times in vehicle journeys (see section 8), a **JourneyPatternInterchange** shall not be used and a **VehicleJourneyInterchange** shall be used instead. **VehicleJourneyInterchange** elements are described in section 9.4.

5.3.6.2   The common interchange group

The common interchange group consists of a number of elements, with permitted values in TXC-PTI as shown in Table 4 below.

Table 4 - Permitted elements in the common interchange group

Element Name		Data Type	Use in TXC-PTI
MinInterchangeTime		*DurationType*	The minimum permitted interchange time (default = 0 mins). Use where appropriate
MaxInterchangeTime		*DurationType*	The maximum time for a guaranteed connection. Use where appropriate.
TransferMode		*AllModesEnumeration*	Not used as assumed that the interchange mode is the mode of the service.
ValidityPeriod	StartDate	*StartDateType*	Not used as assumed that the interchange is valid for the duration of the **OperatingPeriod**.
	EndDate	*EndDateType*	
StoppingArrangements		*NaturalLanguageString*	Not used.
InterchangeActivity		*InterchangeActivityEnumeration*	Mandated in TXC-PTI interchanges. Describes the interchange as follows: *transferOnly*　　not used. *change*　　used for change of vehicle. *through*　　used for remaining on vehicle. *split*　　not used. *join*　　not used.
CrossBorder		*boolean*	Not used.
GuaranteedConnection		*boolean*	Mandated in TXC-PTI interchanges. Set to true for a guaranteed connection. Use for changes of vehicle, or for through connections with the same line name.
ChangeLineNumber		*boolean*	Mandated in TXC-PTI interchanges. Set to true where a through connection changes line name.
Weighting		*integer*	Not used.

5.3.6.3   The journey pattern interchange group

The specific details of the journey pattern interchange are defined within the journey pattern interchange group. There are two elements, **Inbound** and **Outbound**, both of which are mandated by the schema. Each of these is constructed similarly, in that there is a reference to a journey pattern (**JourneyPatternRef**), and the NaPTAN reference (Atcocode) for the stop point at which the interchange takes place in that journey pattern (**StopUsageRef**).

Inbound shall be used for the feeder (incoming) journey pattern, and **Outbound** shall be used for the distributor (outgoing) journey pattern.

5.3.6.4   Examples of a journey pattern interchange

The following examples show how an interchange might be defined at a service level for First in Leicester lines 16, a "pan handle" service, and 17, a circular.

The pan handle line 16 has connections between two different journey patterns in different directions (although reference would need to be made to the journey patterns in question to confirm that this was actually the case). The connection is made at Atcocode 269039050, and for a panhandle service it is clear that this needs to be the same stop on the feeder as the distributor.

```xml
<JourneyPatternInterchange>
    <MinInterchangeTime>PT0M</MinInterchangeTime>
    <InterchangeActivity>through</InterchangeActivity>
    <GuaranteedConnection>true</GuaranteedConnection>
    <Inbound>
        <JourneyPatternRef>JP_12-16-_-y08-1-1-H-1</JourneyPatternRef>
        <StopUsageRef>269039050</StopUsageRef>
    </Inbound>
    <Outbound>
        <JourneyPatternRef>JP_12-16-_-y08-1-5-R-2</JourneyPatternRef>
        <StopUsageRef>269039050</StopUsageRef>
    </Outbound>
</JourneyPatternInterchange>
```

The circular line 17 has connections between the same journey pattern in the same direction (although again this is not clear without reference back to the journey pattern definitions). As can be seen, it is identical in construction to the pan handle interchange.

```xml
<JourneyPatternInterchange>
    <MinInterchangeTime>PT0M</MinInterchangeTime>
    <InterchangeActivity>through</InterchangeActivity>
    <GuaranteedConnection>true</GuaranteedConnection>
    <Inbound>
        <JourneyPatternRef>JP_12-17-_-y08-1-1-H-1</JourneyPatternRef>
        <StopUsageRef>269057024</StopUsageRef>
    </Inbound>
    <Outbound>
        <JourneyPatternRef>JP_12-17-_-y08-1-1-H-1</JourneyPatternRef>
        <StopUsageRef>269057024</StopUsageRef>
    </Outbound>
</JourneyPatternInterchange>
```

As routes (short workings, etc.), timings, and hence journey patterns, can change across the course of the day, or days of the week, there will of necessity be a number of different combinations of interchanges to suit the differing journey pattern combinations.

5.3.7   Other elements

Other elements in **Service** shall be populated as follows:

- **RegisteredOperatorRef** shall be populated, as per the schema, with the (internal) reference of the operator defined in the **Operator** block.

- Where a particular vehicle type with the same set of accessibility features, is commonly used on all trips, **VehicleType** may be populated at **Service** level. This can then be omitted in all **VehicleJourney** elements except those where it is known to be different. Where included, **VehicleType** shall be populated as set out in 2.3 above. As indicated therein, this only indicates the intent of the operator, and is not a guarantee of any particular vehicle actually operating the service as this may change for operational reasons at very short notice.

  Mode shall be populated at **Service** level on the same basis.

- **PublicUse** shall be included and shall state explicitly whether a service is available to the public (true) or whether it is a closed service (false). This allows school services, etc. which are closed to nevertheless be entered into data, and hence allow trips to school to be planned for students. This flag shall be in place of the **SchoolOrWorks** designation, part of the **ServiceClassification**. **ServiceClassification** is related to registrations and is not public-facing, and hence shall not be used within a TXC-PTI document.

Note that a service may be open to the public but have restricted boarding / alighting. For example, the last stop on a public service may be within school grounds first thing in the morning, and only students would be allowed to use it. This restriction is derived from NaPTAN, where a stop may be flagged as private, and users need to reflect both the nature of the service and the nature of the stops when presenting information to the public.

- One or more **Note** elements may be provided, where appropriate. As far as possible, however, all information that is in a note which is encodable as data within a TXC-PTI document should be so encoded. Notes shall be as described in section 2.4.

- **MarketingName** may be included where appropriate. Where a name is simply a long service name, this shall be included in the **Line** element instead. For example, Trent Barton's "black cat" service shall be given that **LineName**. However, the First Norfolk & Suffolk "excel" is a marketing name for a group of lines with names "A", "B", "C", etc.

- **PrivateCode** may be populated, if required by legacy systems.

All other elements not explicitly discussed above shall be omitted in a TXC-PTI document. In particular, this includes the **Description** and **Direction** elements:

- **Description** element was intended to replicate the description box on a registration document, where an operator would describe the change to the service in this registration e.g. "Re-timed journeys in the morning to improve reliability". As TXC has become more widespread, this has often been repurposed to give an overall description of the service e.g. "Hereford – Worcester via Ledbury and Great Malvern". This, though, is unidirectional and there are better ways of achieving the same end as described in 5.5.4 below.

- **Direction** is intended to define the default direction of the service. However, this can lead to confusion and in TXC-PTI the direction will be stated explicitly on each journey patterns.

> In TXC-PTI, **VehicleType**, **Mode**, **Note**, **MarketingName** and **PrivateCode** may be included where appropriate. Where they are, they shall be structured as set out above.

> In TXC-PTI, **RegisteredOperatorRef** and **PublicUse** shall be populated. With the exception of the optional elements listed above and those described in earlier sections, no other elements within **Service** shall be populated.

5.4   The Lines group

The **Lines** group in TXC is a grouping element within a **Service** that can hold one or multiple line records. As noted above a service may consist of one or more related lines, and consequently a **Lines** group in TXC-PTI may also hold one or multiple line records.

As well as the use case outlined in 5.1 above, an operator may also operate two distinct patterns of operation which are seasonal in nature, i.e. a summer and winter timetable. Where these do not vary from year to year, the operator would not wish to continually update the data to swap between the two and would wish instead to have both patterns within the same TXC-PTI document.

This type of arrangement could very easily be handled through individual dated **VehicleJourney** records. However, this doesn't give a clear separation of the timetables and analysis of the individual records would need to be carried to understand, how to present the "summer" and "winter" timetables to the public.

For these situations, then, it is preferable to have the two patterns coded as distinct lines, albeit sharing the same **LineName**, within TXC-PTI. It is then clear through the referencing which pattern is which, and which trips belong with which pattern.

> For TXC-PTI, a **Lines** group may contain multiple **Line** elements, so long as the lines are related either spatially (e.g. 1/1A) or temporally (e.g. summer & winter timetables on line 1).

5.5   The Line element

5.5.1   Introduction

The **Line** element in TXC describes one of the lines that comprise the service, provides a unique identity for the line, and describes how the line shall be known to the public. It also specifies the separate inbound and outbound descriptions of the service, e.g. outbound as "Hereford to Worcester via Ledbury and Great Malvern", inbound as "Worcester to Hereford via Great Malvern and Ledbury".

A **Line** shall be versioned, as described in 0 above.

As stated earlier, there may be multiple **Line** records in a single TXC-PTI document. However, there shall only ever be one version of a given line in a single document.
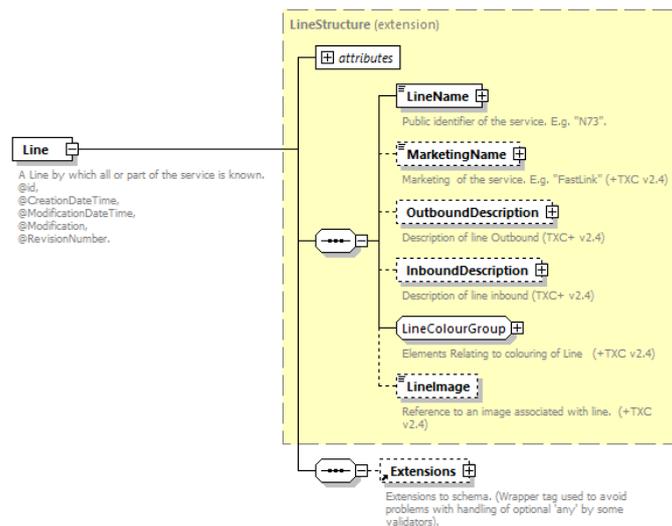
Figure 14 - The Line element

5.5.2   Identifying the line

The *id* attribute of the **Line** element uniquely identifies the line. For TXC-PTI this shall take the form of a number of different components, separated by colons, as follows:

- The most granular NOC for the operator (e.g. FESX for First Group's bus operations in Essex);

- The **ServiceCode** (to assist in linking to other lines in the same group of lines); and

- The line name.

For the avoidance of doubt, the line *id* shall not include a version number. Version control shall be in accordance with 0 above.

For example, two distinct line 8s operated by First in Essex might appear (in separate TXC-PTI documents, within their own individual **Service** elements) as:

```xml
<Service RevisionNumber="1">
   <ServiceCode>PF0000599:41</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:41:8">
         <LineName>8</LineName>
      </Line>
   <Lines>
   ...
   <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
   ...
</Services>
```

and

```xml
<Service RevisionNumber="4">
   <ServiceCode>PF0000599:75</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:75:8">
         <LineName>8</LineName>
      </Line>
   <Lines>
   ...
   <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
   ...
</Services>
```

Where a service contains two lines which are the seasonal variants of the same service, this can be indicated by the addition of an additional seasonal identifier at the end of the line *id*. This should be separated from the rest of the id by a colon.

For example, if the service 8 in the example above had summer and winter timetables, then these could be identified as follows:

```xml
<Service RevisionNumber="4">
   <ServiceCode>PF0000599:75</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:75:8:winter">
         <LineName>8</LineName>
      </Line>
      <Line id=" FESX:PF0000599:75:8:summer">
         <LineName>8</LineName>
      </Line>
   <Lines>
   ...
   <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
   ...
</Services>
```

Note that some services may consist of lines that operate all year round, with additional summer trips on a different line number. In this case it is acceptable in TXC-PTI to place the seasonal identifier on the end of the seasonal line *id*, even though there is no corresponding winter timetable.

Note also that such timetables will, in any event, need to be refreshed at least once a year in order to put the correct dates of operation for that year into the service.

Seasonal timetables are the only instance in TXC-PTI where a line may be split. It is not permitted, for example, to have separate TXC-PTI lines or documents for e.g. Mondays to Fridays, Saturdays, etc. operations.

In TXC-PTI, the line *id* shall follow a consistent format of the NOC, the ServiceCode, the line name, and, if appropriate, the seasonal identifier, separated by colons. It shall not include a version number.

5.5.3 Line name and marketing name

The **LineName** element is mandatory and shall contain the public-facing name of the service, in full. For example, Trent Barton's "indigo" service shall have a **LineName** of "indigo", and not any shortened version of it (e.g. "IGO").

The **MarketingName** is optional and shall be included where a line has a marketing name that is different to other lines within the service. Otherwise, **MarketingName** shall be omitted and any marketing name included at the **Service** level.

5.5.4 Describing the line

In a TXC-PTI document, the line shall have a description for each direction of the line that describes the overall route of the line in general terms. It does not have to describe every place served but should be sufficient to identify the line in public-facing publicity.

This is achieved by use of a line's **OutboundDescription** and **InboundDescription** elements. Both of these are constructed the same way, using a LineDescriptionStructure as shown below:
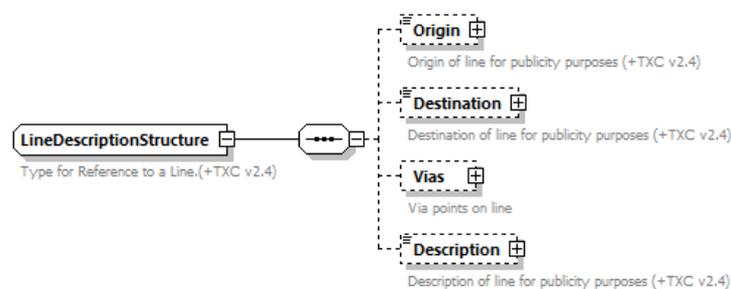
Figure 15 - The line description structure

This structure allows the description to be specified granularly, using **Origin**, **Destination**, and a set of some or all of the **Vias** (listed in order of calling), and these elements can be populated in TXC-PTI if required. The **Description** field, however, is mandatory and shall contain a pre-formatted textual description of the inbound and outbound directions. The earlier example showed, for instance, that a line might be described as "Hereford to Worcester via Ledbury and Great Malvern" outbound, and as "Worcester to Hereford via Great Malvern and Ledbury" inbound.

Some general points to note, for TXC-PTI:

- For each direction of the line, there shall be a corresponding description.

- If any direction is not present in the line, the corresponding description shall be omitted.

- There are additional directions that are possible for a line, including *clockwise* and *anti-clockwise*. For the purposed of TXC-PTI, the **OutboundDescription** shall be used for *clockwise* directions, and the **InboundDescription** shall be used for *anti-clockwise* directions.

> In TXC-PTI, each direction of a line shall have a matching description element that describes the route of that particular line in that direction. Clockwise and anti-clockwise services are mapped to outbound and inbound description elements respectively.

6   Stop points and stop areas

6.1   Introduction

In a TXC document, stop point declarations are required so that they can be referenced, ultimately, in the individual vehicle journeys. There are two ways provided in TXC for doing this:

a)   by referencing a stop already in NaPTAN; or

b)   by creating a new stop record in the same format as a stop record in NaPTAN.

In addition, TXC allows the creation of NaPTAN "stop areas" to replace or supplement those in NaPTAN.
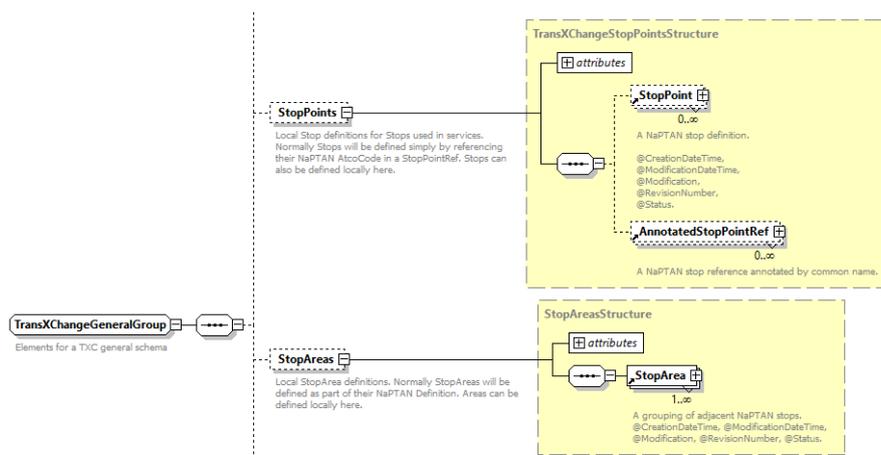
Figure 16 - StopPoints and StopAreas in a TXC document

The presumption in a TXC-PTI document is that it shall always prefer to use a reference to a NaPTAN stop record, and that the creation of new stop records shall be reserved for the very specific cases set out below and shall be replaced by a NaPTAN reference as soon as one exists. Additionally, there is little or no merit in creating a **StopArea** within a TXC document, so **StopAreas** shall be omitted in TXC-PTI.

> In TXC-PTI, **StopPoints** shall use **AnnotatedStopPointRef** elements as far as possible to refer back to NaPTAN, with **StopPoint** being used sparingly and only for specific use-cases. **StopAreas** shall be omitted.

6.2   Stop points

As already indicated, the presumption in TXC-PTI is that reference is made to an existing NaPTAN stop, using an **AnnotatedStopPointRef**, which has the structure shown below.

Of the elements that are included within the **AnnotatedStopPointRef**, only two are required by the schema: **StopPointRef**, which is the actual NaPTAN reference (Atcocode) of the NaPTAN stop; and **CommonName**, which is intended to be a re-statement of the name of the stop in NaPTAN in order to aid readability and diagnostics in a TXC file. Other optional elements, which were extended in TXC v2.4, contain more of the information from the NaPTAN file.
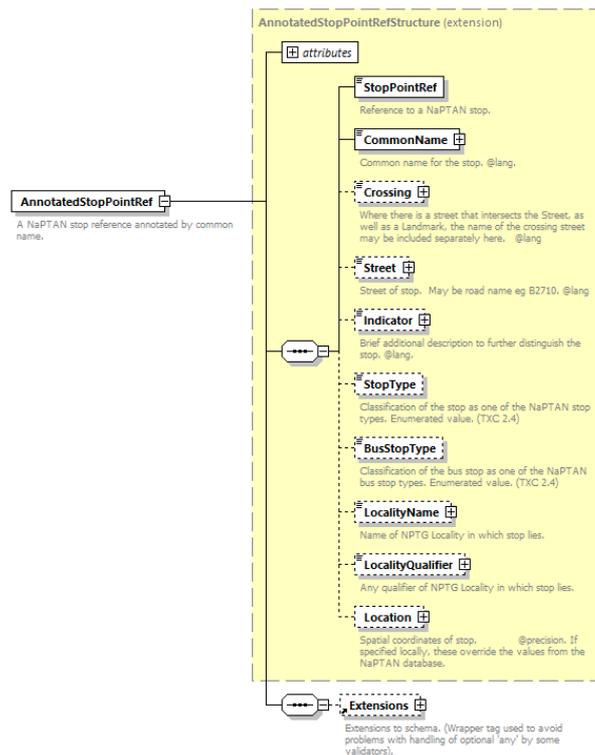
Figure 17 - The AnnotatedStopPointRef element

At no point was it intended that the elements populated in an **AnnotatedStopPointRef** structure should replace or overwrite the contents of NaPTAN.

This intent is continued, and made explicit, in TXC-PTI. **StopPointRef** is essential, and **CommonName** is mandated by the schema, so both shall be present in a TXC-PTI document.

Additionally, to aid readability, **Indicator**, **LocalityName** and **LocalityQualifier** may be included in a TXC-PTI document. Other elements shall be omitted and taken from NaPTAN.

In general, where a stop point field in TXC-PTI is different to NaPTAN, then the details held in NaPTAN shall be used in preference. If the naming in NaPTAN is deficient then the naming in NaPTAN should be discussed with the stop owner and agreement reached as to the accepted name.

Note that NaPTAN is a granular schema, with the different parts of the naming broken down into different fields. This allows users to compose stop names that suit their purposes from the different fields, which may be different depending on usage. For example, an express service between two towns, which only calls once in each locality, may well refer to the stop by just the locality name; conversely, a service which runs solely in one of those towns may well call the same stop by its common name only, and omit the locality completely. This is up to the end user, and data should not be overwritten simply to accommodate one use-case.

Any stop point that is used shall be appropriate to the service being described. For example, a bus service shall use only bus stops (either on-street or in bus stations) and shall not use rail station or ferry terminal stops, taxi ranks, etc. Likewise, where a TXC-PTI document is being used for other modes, those modes shall also use appropriate stops – metro service shall use metro stops, etc.

In TXC-PTI, an **AnnotatedStopPointRef** shall be used to refer to an appropriate NaPTAN stop. It shall contain a **StopPointRef** and a **CommonName**, and may optionally include an Indicator, a **LocalityName** and a **LocalityQualifier**. It shall not have other elements.

As indicated earlier, there will be occasions where an operator needs to refer to a stop that is not yet in NaPTAN, or which has no real need to be in NaPTAN e.g. temporary stops for festivals, etc. In such cases, which will be set out in legislation and guidance, an operator may declare a new stop using a **StopPoint** element. This element shall be populated according to the rules set out in NaPTAN, the detail of which is outside of the scope of this document. Users should refer to the NaPTAN schema guidance document.

7   Routes and tracks

In TXC, a **Route**, together with its constituent **RouteSections** and **RouteLinks**, describes the physical sequence of stopping points served. At its most basic it is a series of NaPTAN stop points and nothing more, which therefore provides a routing which is independent of any particular coordinate system. A particular route can be projected onto the underlying road network through the use of the **Tracks**. The following diagram from a route in Exeter serves to illustrate this:
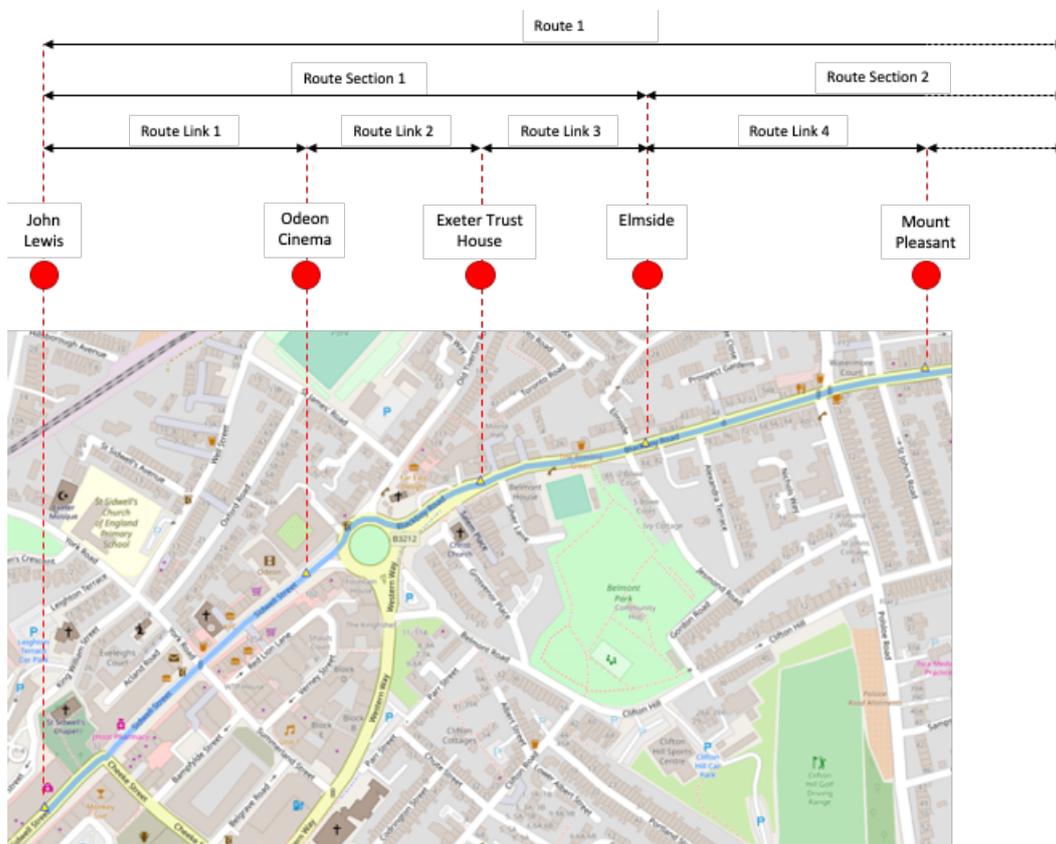
Figure 18 – Example of routes and tracks showing different levels

It is important to note that, within a TXC document, a route does not have any timing information associated with it. It is simply a series of stops which may be described in geographic terms by the addition of geographic points in a track. Because of this, and because a journey pattern restates the links that occur, a **Route** is an optional element in TXC, and several systems omit it.

Omitting track information, however, is limiting when the data will be used to drive real-time systems; prediction engines work best if they understand the route distance between stops as accurately as possible. Without track information, such systems will only be able to use "crow flies" distances between stops derived from stop coordinates in NaPTAN, or they will have to add information or estimate the tracks themselves. This is clearly undesirable, and hence **Route**, **RouteSection**, and **RouteLink** elements are mandated in TXC-PTI documents, with **Track** elements being strongly recommended.

These elements shall be populated in accordance with the TXC schema. Element ids can be constructed to suit the originating system, and there are no specific rules for TXC-PTI except that:

- The **RouteLink/Distance** element may be populated as an intermediate stage between no track and full track, although is shall be populated with an accurate measurement of the travelled distance between the two stops.

- The **RouteLink/Direction** element shall not be populated, in order to facilitate re-use of definitions between different routes without potential conflicting directions.

Additionally, a tendency in TXC is to export entire routes as single route sections, which then repeat sequences of stops when there are variations to routes e.g. short workings. It is strongly recommended that a TXC-PTI document uses efficiency when creating **RouteSections**, breaking them down into logical sections, so that re-use within **Routes** is readily achievable. This will help limit file sizes.

It is noted that operators submitting EBSR documents are required to provide tracks which have at least one point on every individual street that the vehicle travels down, in sequence. Where such information exists then this minimum information shall be used to populate a **Track**, although operators should strive to get full routing information into TXC-PTI as soon as practical.

In TXC-PTI, **Route**, **RouteSection**, and **Route** elements shall be provided. It is strongly recommended that **RouteSections** are logically structured to facilitate re-use within **Routes** to help minimise file sizes.

In TXC-PTI, **Track** elements are strongly recommended. Where possible, operators should populate these with, at a minimum, the single point on each street as per the EBSR requirements.

8   Journey timings on standard services

8.1   Introduction

Journey timings in TXC are handled through **JourneyPatterns**, **JourneyPatternSections** and **JourneyPatternTimingLinks**. These relate structurally to the non-timed equivalents **Routes**, **RouteSections** and **RouteLinks**. TXC permits a **JourneyPattern** to operate over a shortened version of the **Route** that it maps onto, but for TXC-PTI this is not recommended and a **JourneyPattern** (and its sub-elements) shall have direct one-to-one mapping to the underlying **Route** (and its sub-elements).

As with routes, there has been a tendency in TXC to generate journey patterns which consist of a single journey pattern section with all of the timing links required for that particular journey pattern. Like routes, this leads to a significant increase in the file size since there is a high degree of redundancy. Therefore, TXC-PTI recommends that **JourneyPatterns**, like **Routes**, shall be structured efficiently so that sectional re-use is facilitated. This requirement in any case derives from the requirement to map journey patterns onto routes with a one-to-one mapping.

In TXC-PTI there shall be at least one **JourneyPattern** for each direction of a non-flexible service (**StandardService** – see section 5.3.5), and this (and its sub-elements) shall have a one-to-one mapping to a **Route** (and its sub-elements).

In TXC-PTI it is strongly recommended that **JourneyPatternSections** are logically structured to facilitate re-use within **JourneyPatterns** to help minimise file sizes.

Note that TXC is a link-based schema. That means that a vehicle journey states the departure time of the trip, but that thereafter there is a duration over each successive link. This avoids problems where times can occur out of sequence in call-time based systems, but does mean that the actual call times at each stop need to be calculated by the consuming system by summing the link durations and any wait times up to that point and adding this to the departure time.

In TXC, there are two common ways of populating **JourneyPatternSections** (and the **JourneyPatternTimingLinks** that they contain):

1) There is a **JourneyPattern**, **JourneyPatternSection** and **JourneyPatternTimingLink** set for every different combination of link duration, wait time and stop activities, with each piece of information explicitly written in. Individual vehicle journeys reference the appropriate journey pattern; and

2) There is a single **JourneyPattern**, **JourneyPatternSection** and **JourneyPatternTimingLink** set for each direction, with each link duration and wait time set to zero minutes. This method uses the facilities in the vehicle journey to override the default zero-minute durations for each of the links. Stop activity, however, is usually included in the **JourneyPatternTimingLink** in the same way as method (1).

TXC-PTI will accept either way of defining the timings, so long as:

*   only one method is used, not a mix of methods; and

*   all links in method 2 are set to zero minutes duration, and all vehicle journeys explicitly state the duration and wait time for each link.

For the avoidance of doubt, TXC-PTI permits the definition of the stop activity within the **JourneyPatternTimingLink**, as set out below, which shall be inherited by the vehicle journeys even when link durations are being overridden.

In TXC-PTI, either sufficient **JourneyPatternSection** elements shall exist to fully define the different timing and activities of a service, with all values explicitly stated, or there shall be a minimum of **JourneyPatternSection** elements where all of the durations and wait times are set to zero.

8.2   The JourneyPattern element

A **JourneyPattern** element is constructed using a *JourneyPatternStructure*, as shown below.
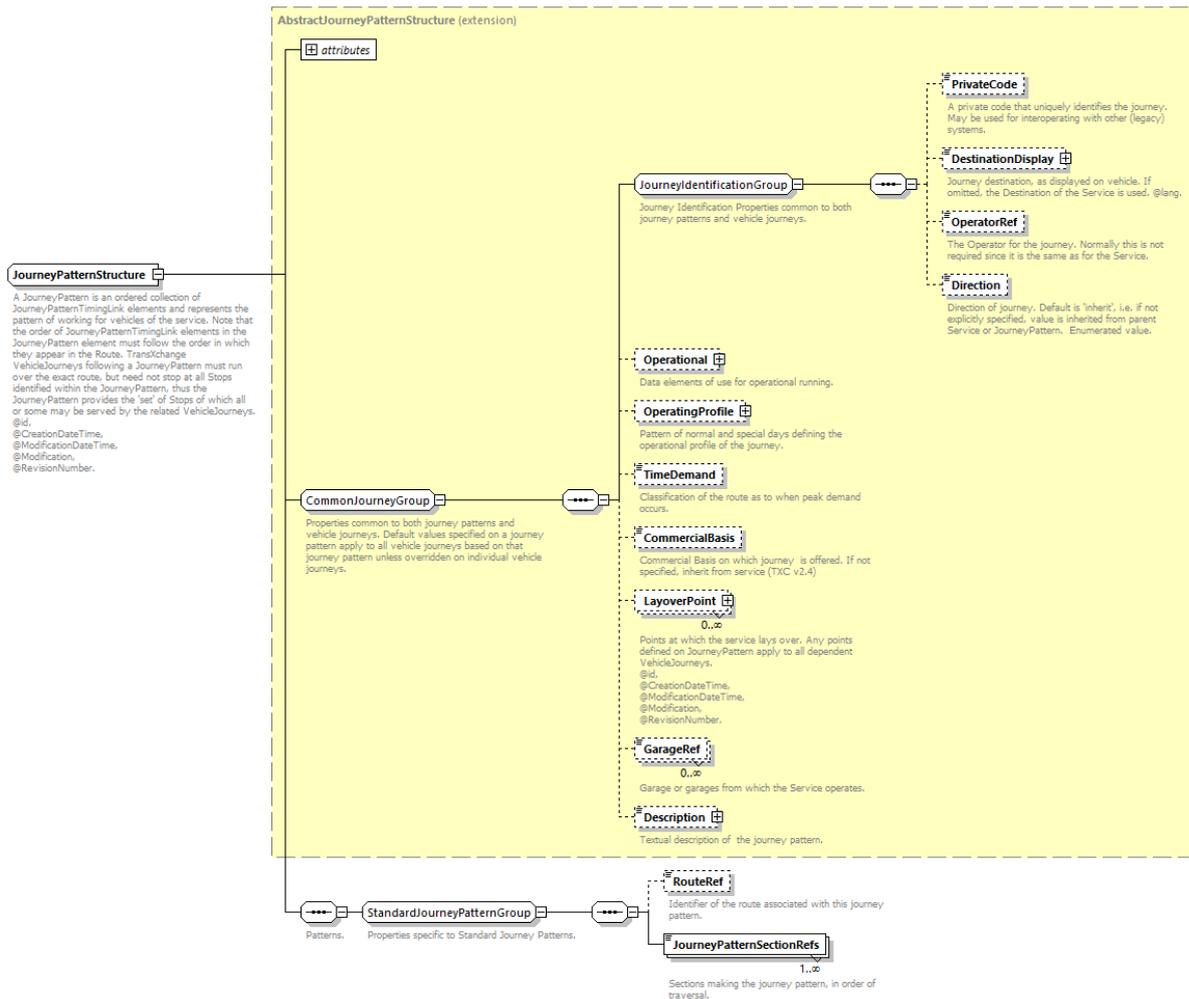
Figure 19 - The JourneyPattern structure

For TXC-PTI, this shall be populated as follows:

- A **PrivateCode** may be included for compatibility with legacy systems.

- A **DestinationDisplay** element shall be provided unless a DynamicDestinationDisplay is provided for all stops in the **JourneyPatternTimingLink**. This allows systems to state the ultimate destination of the journey.

- An **OperatorRef** shall be included and shall reference the *id* of the (single) **Operator** element.

- **Direction** shall be included at the **JourneyPattern** level. It shall then be assumed that all **RouteSections** and **RouteLinks** are in the same direction. This allows re-use without confusion of, for example, route sections describing a route through a town centre which is used in multiple directions.

- **Operational** and **OperatingProfile** elements shall not be provided at the **JourneyPattern** level. These are best placed within individual vehicle journeys. An **OperatingProfile** default may additionally be provided at the **Service** level.

- A **RouteRef** shall be provided to link the journey pattern to the corresponding route.

- A series of **JourneyPatternSectionRefs** shall be provided, referencing the sequence of journey pattern sections that together make up the journey pattern. As stated elsewhere,

the journey pattern and its associated elements should be structured so as to facilitate re-use of journey pattern sections. There shall therefore, ideally, be multiple **JourneyPatternSectionRefs** in any given **JourneyPattern**.

- Other elements shall be omitted as they contain no useful public-facing information.

8.3   The JourneyPatternSection element

A **JourneyPatternSection** is made up of one or more **JourneyPatternTiming** link elements, in sequence. There are no additional constraints placed upon a **JourneyPatternSection** by TXC-PTI other than those already set out with regards to mapping to **RouteSection** elements and to structuring sections to facilitate re-use by routes.

The *id* of a **JourneyPatternSection** can be constructed to suit the originating system.

8.4   The JourneyPatternTimingLink element
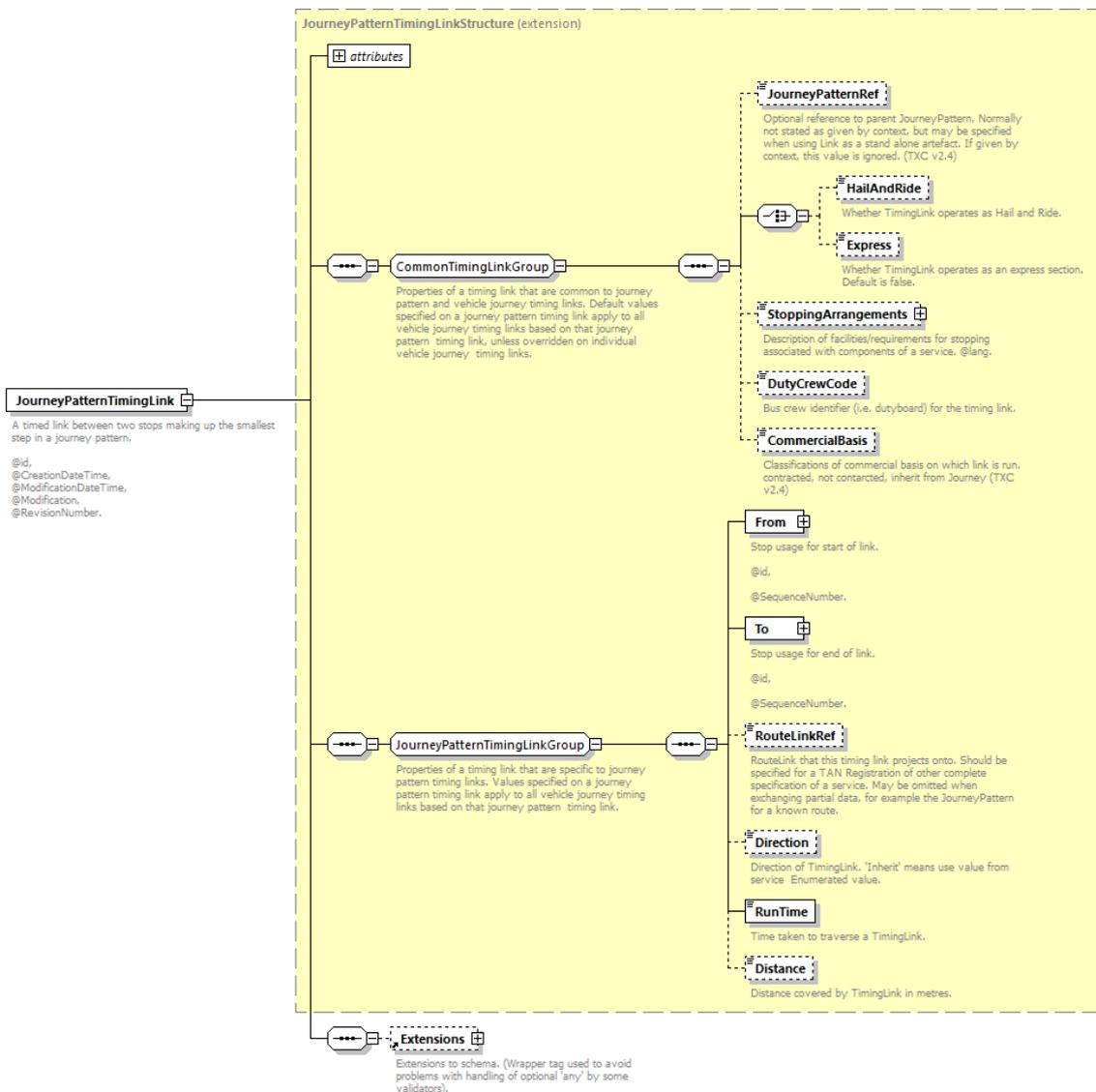
Figure 20 - Overview of JourneyPatternTimingLink element

8.4.1 Introduction

The **JourneyPatternTimingLink** element contains the details of the timing of the stop in the sequence, and also the details of the activity that is carried out at the stop: whether passengers are allowed to board and alight; whether or not there is a dwell time at the stop; etc.

> In TXC-PTI, a **JourneyPattern** shall have an **OperatorRef**, **Direction**, **RouteRef** and one or more **JourneyPatternSectionRefs**. It shall also have a **DestinationDisplay** if a **DynamicDestinationDisplay** is not provided elsewhere. It shall omit other elements.

8.4.2 Basic Structure of a JourneyPatternTimingLink

For TXC-PTI, a **JourneyPatternTimingLink** shall be populated as follows:

- **RouteLinkRef** shall be populated with the id of the corresponding **RouteLink**.

- **RunTime** shall be populated either with the actual value of the run time over the link, or with a duration of zero minutes, as described in section 8.1.

- **Distance** may be populated if available. If it is populated, then it shall have the same value as the **Distance** element on the corresponding **RouteLink**.

- **DutyCrewCode** shall be omitted, as this is better placed in individual vehicle journeys.

- **Direction** shall be omitted from a **JourneyPatternTimingLink**. This permits reuse of sections and links where, for example, a section of a service is used in two different directions such as a loop through a town centre due to one-way systems.

- **JourneyPatternRef**, **HailAndRide**, **Express**, **StoppingArrangements** and **CommercialBasis** shall be omitted. The reference to the journey pattern is implied, and the other elements do not contain useful public-facing information. There might be some merit in describing a link as **HailAndRide**, but this would require detailed track information which is only recommended rather than mandatory, and it would have to apply to the whole of the link even if only part of it was hail and ride.

> In TXC-PTI, a **JourneyPatternTimingLink** shall have a **RouteLinkRef** and **RunTime**, and may have a **Distance**. It shall additionally have **From** and **To** elements but shall not have other elements. Specifically, it shall not have a **Direction** element.

8.4.3 Stop definition and wait times

In addition to this, every link requires a **From** element, detailing the departure stop and activity, and a **To** element, detailing the arrival stop and activity. These two elements are constructed the same way, using a *JourneyPatternStopUsageStructure*.
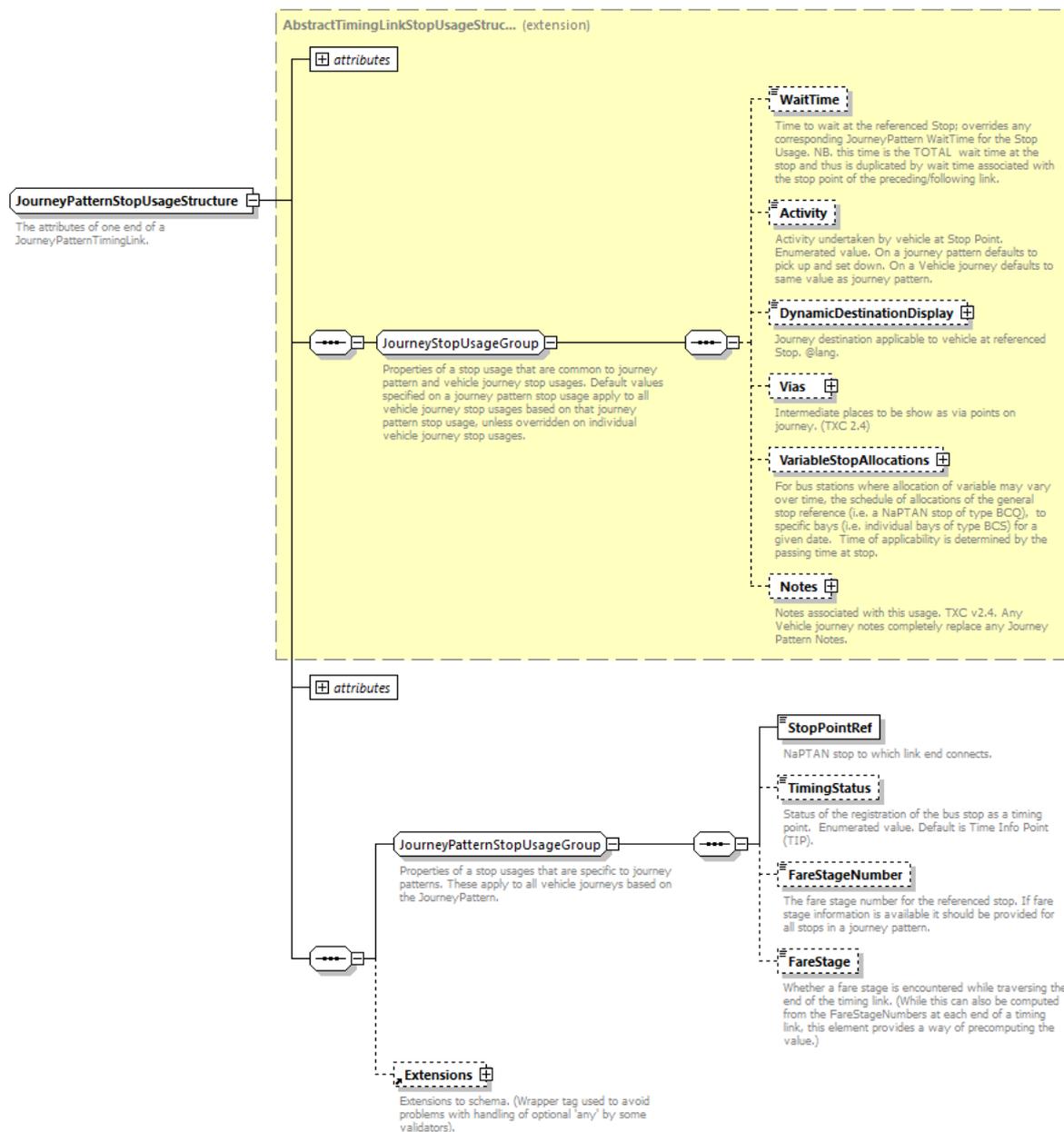
Figure 21 - The JourneyPattern stop usage structure

In versions of TXC prior to v2.4, the intent of these structures was that they were additive. That is, they combined to give an overall effect for activity at the stop, remembering that the arrival stop (**To** element) on one link is the same physical stop point as the departure stop (**From** element) on the next successive link. So, for example, if a **To** element defined a stop as "set down only" and a **From** element defined a stop as "pick up only" then the overall effect was that it allowed both setting down and picking up.

One area where this caused particular confusion was with wait times. Both arrival and departure settings could define a wait time – the intention was, originally, to be able to specify an unloading time and a loading time separately, although in practice this never happened.

In TXC v2.4, this behaviour fundamentally changed. The schema guidance, unfortunately, did not reflect this, but it is clear from the schema. From version 2.4 onwards, the wait time was to be the total wait time and was to be stated identically on both the **To** element of the incoming link and the **From** element of the next, outgoing link.

TXC-PTI is making this requirement explicit. Moreover, there are additional benefits to making as many of the other elements as possible behave the same way, as this aids readability and is unequivocal as to what is happening at the stop. Some elements still cannot be treated this way, however, and these are outlined below.

A **JourneyPattern** in TXC-PTI shall therefore use the elements and values shown in the table below.

Table 5 - Permitted elements and values in a TXC-PTI JourneyPattern

Element Name	Data Type	Use in TXC-PTI
WaitTime	*DurationType*	The total wait time at the stop, which is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link. Shall be included where appropriate.
Activity	*VehicleAtStopActivityEnumeration*	The activity at the stop, which is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link. Shall always be included. Permitted values are *pickup*, *setDown*, *pickUpAndSetDown* and *pass*. Note that, as information is to be duplicated, the majority of stops will be shown as *pickUpAndSetDown.*
DynamicDestinationDisplay	*NaturalLanguageString*	A textual description of the destination of the vehicle which may be shown at this stop. This is different from the overall destination of the journey pattern and can vary along the route. It is particularly useful for circular services where, for example, the destination shown at the first stop would otherwise be the same stop. Shall always be included if **JourneyPattern/DestinationDisplay** is not included and shall be included if appropriate otherwise.
Vias	*ViasStructure*	Not used.
VariableStopAllocations	*VariableStopAllocationsStructure*	Not used.
Notes	*NotesStructure*	May be included where appropriate. See section 2.4. May be different on the **To** end of the incoming link and the **From** end of the next, outgoing link.
StopPointRef	*StopPointRefStructure*	The AtcoCode of the NaPTAN stop point, or the pseudo-AtcoCode of the operator-defined stop point. It is necessarily duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link as it is the same stop.

Element Name	Data Type	Use in TXC-PTI
TimingStatus	*TimingStatusEnumeration*	Defines the reliability of the timing at the stop and may also be used as a flag to aid in the display of "main point" timetables. Shall always be included. TXC v2.4 includes both the older three-letter codes as well as newer, more readable versions. To aid readability, only those codes set out below shall be used in a TXC-PTI document. Allowed values are: *principalTimingPoint* (note: only this spelling) *otherPoint*
FareStageNumber	*FareStageNumberType*	States the fare stage number for this stop. May be different on the **To** end of the incoming link and the **From** end of the next, outgoing link. May be included where appropriate.
FareStage	*boolean*	A true/false flag which provides a simple way to determine if a fare stage is changing at this stop. It is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link.

In TXC-PTI, a **From** element and the **To** element on the next link shall have the same information for the stop except where stated otherwise.

8.4.4   Formatting for timetable displays

One of the common problems with TXC data is how disparate trips on different days of the week can be structured into a traditional matrix timetable, and one of the goals of the TXC-PTI profile was to facilitate this where possible.

To achieve this, a TXC-PTI document shall use the *sequenceNumber* attribute of a **JourneyPatternTimingLink**.

The sequence number is not, as has sometimes been seen, simply a number which represents the order of links in a particular **JourneyPatternSection**. Rather, it is a number which represents that stop's position in a row on a timetable. This is best illustrated by considering the following timetable:

Table 6 - Example of a structured timetable

Seq. No.	Name	AtcoCode	Trip 1	Trip 2
1	One	1580ABCD	08:00	08:15
2	Two	1580EFGH	08:02	08:17
3	Three	1580JKLM	|	08:23
4	Four	1580NPQR	08:07	-

In TXC-PTI, this shall be coded in this way:

```xml
<JourneyPatternSection id="ABC1_jps1">
   <JourneyPatternTimingLink id="ABC1_jps1_tl1">
      <From SequenceNumber="1">
         <StopPointRef>1580ABCD</StopPointRef>
      </From>
      <To SequenceNumber="1">
         <StopPointRef>1580EFGH</StopPointRef>
      </To>
      <RunTime>PT2M</RunTime>
   </JourneyPatternTimingLink>
   <JourneyPatternTimingLink id="ABC1_jps1_tl2">
      <From SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </From>
      <To SequenceNumber="4">
         <StopPointRef>1580NPQR</StopPointRef>
      </To>
      <RunTime>PT5M</RunTime>
   </JourneyPatternTimingLink>
</JourneyPatternSection>
<JourneyPatternSection id="ABC1_jps2">
   <JourneyPatternTimingLink id="ABC1_jps2_tl1">
      <From SequenceNumber="1">
         <StopPointRef>1580ABCD</StopPointRef>
      </From>
      <To SequenceNumber="1">
         <StopPointRef>1580EFGH</StopPointRef>
      </To>
      <RunTime>PT2M</RunTime>
   </JourneyPatternTimingLink>
   <JourneyPatternTimingLink id="ABC1_jps2_tl2">
      <From SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </From>
      <To SequenceNumber="3">
         <StopPointRef>1580JKLM</StopPointRef>
      </To>
      <RunTime>PT6M</RunTime>
   </JourneyPatternTimingLink>
</JourneyPatternSection>
```

Note the *SequenceNumber* usage in the **From** and **To** elements; it is clear from the numbering used which order the stops should appear, and which times shall appear against which stops.

In TXC-PTI, a **JourneyPatternTimingLink** shall use the *SequenceNumber* attribute in the **From** and **To** elements in order to facilitate the correct layout of a timetable.

9   Vehicle journeys

9.1   Introduction

In TXC, the **VehicleJourney** element defines the actual trip that a vehicle will make. Up to this point, the TXC has been defining the basic parameters of the service, the routings of the vehicles, the default timings of the links, etc., but this is the section of the TXC document where the actual departures are specified. It is essential that this information is correctly structured and reflects what the vehicle will do.

TXC-PTI adopts the approach that the best place to hold the information about a trip is within the trip itself. To that end it recommends that much of the detail about the trip, and in particular the

dates on which the trip operates, be contained within the **VehicleJourney** and not elsewhere. However, for reasons of readability and compactness it is possible to define default patterns and timings that suit the majority of the trips, and then override them in individual vehicle journeys where appropriate. In this case, though, complete sets of information shall be overridden and the systems shall not rely on inheriting only parts of data elements as this leads to confusion and errors of interpretation.

9.2   Basic structure of a vehicle journey
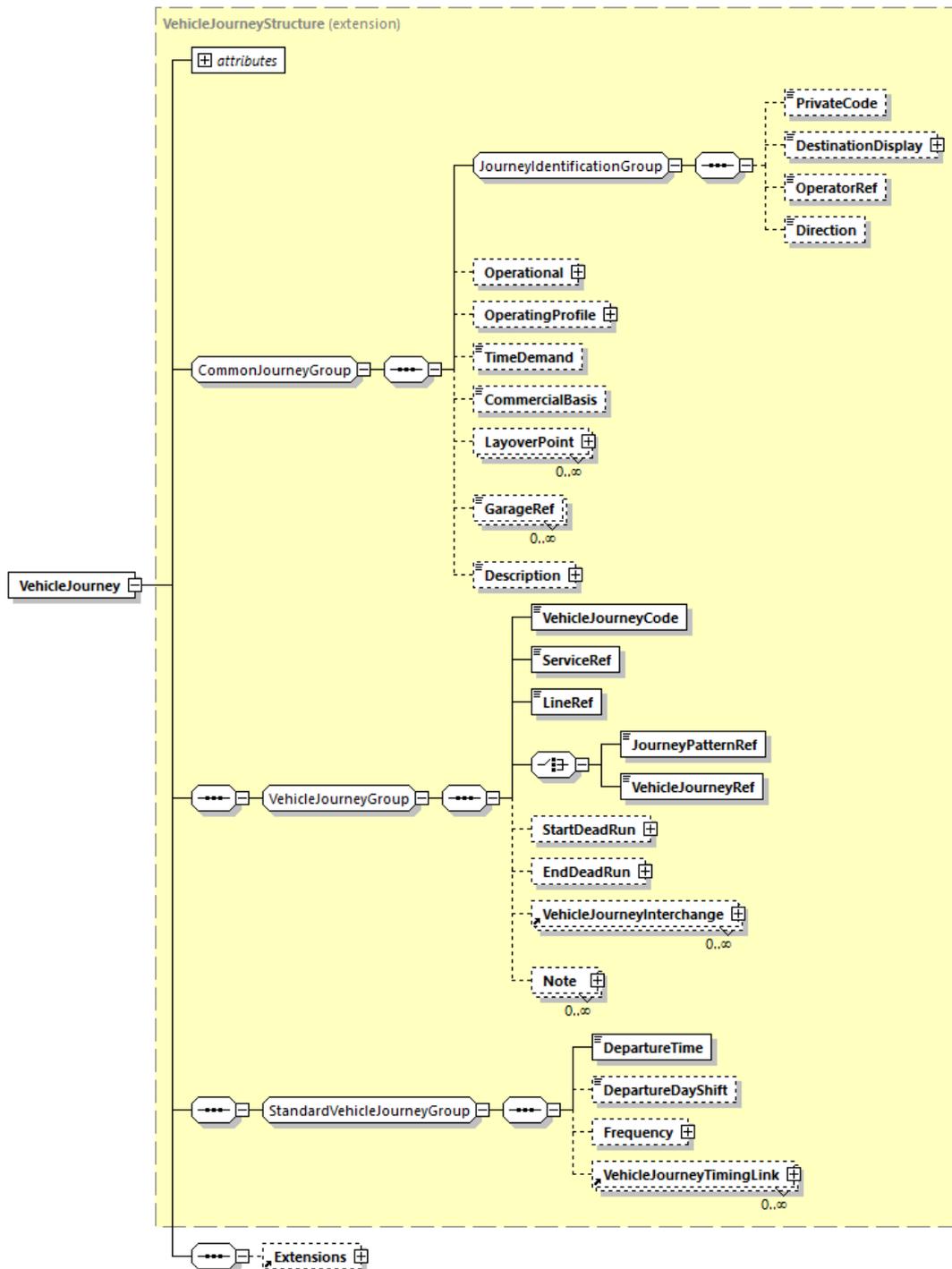
9.2.1   Introduction

Figure 22 - Basic structure of a VehicleJourney

A vehicle journey record is structured as shown in the figure above. Many of these elements are described in more detail in the sections that follow. In general terms, however, a **VehicleJourney** in TXC-PTI:

- shall have a **VehicleJourneyCode**, a **ServiceRef** and a **LineRef** as mandated by the schema. TXC-PTI does not specify how a **VehicleJourneyCode** is to be structured, and this can be done to suit the originating system. It may however be useful to include the line ref as part of the code.

- shall have a **DestinationDisplay** where neither this nor a **DynamicDestinationDisplay** is provided at a **JourneyPattern** level.

- shall have either a **JourneyPatternRef**, referencing a **JourneyPattern**, or a **VehicleJourneyRef**, referencing another **VehicleJourney**. If a **VehicleJourneyRef** is used, however, then all the details of that **VehicleJourney**, including operating profile, timings, vehicle types, etc. shall be inherited and not overridden. The only elements that may be amended are the operational elements for real time systems.

- shall have a **DepartureTime** to say what time the **VehicleJourney** starts, and may optionally shift trips to the next day using **DepartureDayShift** (see section 9.5) or define them as frequent journeys using **Frequency** (refer to TXC schema guidance for specifying frequent journeys).

- may override the operating days by providing a complete set of days and dates when the trip operates using an OperatingProfile block (section 9.3)

- may override the vehicle timings by providing a complete set of times over all of the links in a journey pattern in a set of **VehicleJourneyTimingLink** elements (see section 9.4).

- shall have an **Operational** block and set of **GarageRef** elements containing information to support real time systems, where such information is available.

- may include dead runs (as **StartDeadRun** and **EndDeadRun**) where appropriate (see section 9.7).

- may have a **PrivateCode** for compatibility with legacy systems.

- may include information on interchanges where appropriate (see section 9.6).

- shall not use **TimeDemand**, **CommercialBasis**, **LayoverPoint**, or **Description** as these contain no useful public-facing information.

> In TXC-PTI, a **VehicleJourney** shall define the operation of a particular trip, optionally overriding some of the blocks so long as these are complete replacements. Some elements are not to be used.

9.2.2   Formatting for timetable displays

As with stops read down a timetable, one of the common problems with TXC data is how disparate trips on different days of the week can be structured across a traditional matrix timetable, and one of the goals of the TXC-PTI profile was to facilitate this where possible.

To achieve this, a TXC-PTI document may use the *sequenceNumber* attribute of a **VehicleJourney**.

The sequence number is not, as has sometimes been seen, simply a number which represents the order of the records in a particular set of **VehicleJourneys**. Rather, it is a number which represents

that trip's position in a column on a timetable. This is best illustrated by considering the same timetable as before:

Table 7 - Example of a structured timetable

Seq. No.	Name	AtcoCode	Trip 1	Trip 2
1	One	1580ABCD	08:00	08:15
2	Two	1580EFGH	08:02	08:17
3	Three	1580JKLM	|	08:23
4	Four	1580NPQR	08:07	-

In TXC-PTI, this shall be coded in this way:

```
<VehicleJourneys>
   <VehicleJourney Sequenceumber="1" >
      <VehicleJourneyCode>Trip_1</VehicleJourneyCode>
      ...
   </VehicleJourney>
   <VehicleJourney SequenceNumber="2">
      <VehicleJourneyCode>Trip_2</VehicleJourneyCode>
      ...
   </VehicleJourney>
   ...
</VehicleJourneys>
```

Note the *SequenceNumber* usage in the **VehicleJourney** elements; it is clear from the numbering used which order the trips should appear. However, care does have to be taken when creating sequence numbers to give some thought as to how the trips are to be presented, because the order can vary depending on how trips are to be combined – for example, on a "Mondays to Saturdays" timetable compared to separate "Mondays to Fridays" and "Saturdays" timetables. For this reason, serquence numbers on vehicle journeys are optional in TXC-PTI, although they are recommended.

In TXC-PTI, a **VehicleJourney** may use the *SequenceNumber* in order to facilitate the correct layout of a timetable. This is strongly recommended.

9.3   The operational profile

9.3.1   Basic structure of an operational profile

An operational profile details the regular and irregular operation of a trip.

It consists of a *NormalDayPropertiesGroup*, which defines the regular operation, broken down into regular days which are then qualified (restricted) by periodic elements (e.g. 1st Wednesday of each month) or by serviced organisation dates (such as school term dates).

In addition to the regular days, elements also exist to add or remove specific date ranges, and to add or remove specific bank holidays (either enumerated or user-defined). Each of these has days of operation, which are in addition to any defined regular days, and days of non-operation, which remove any previously defined regular days.

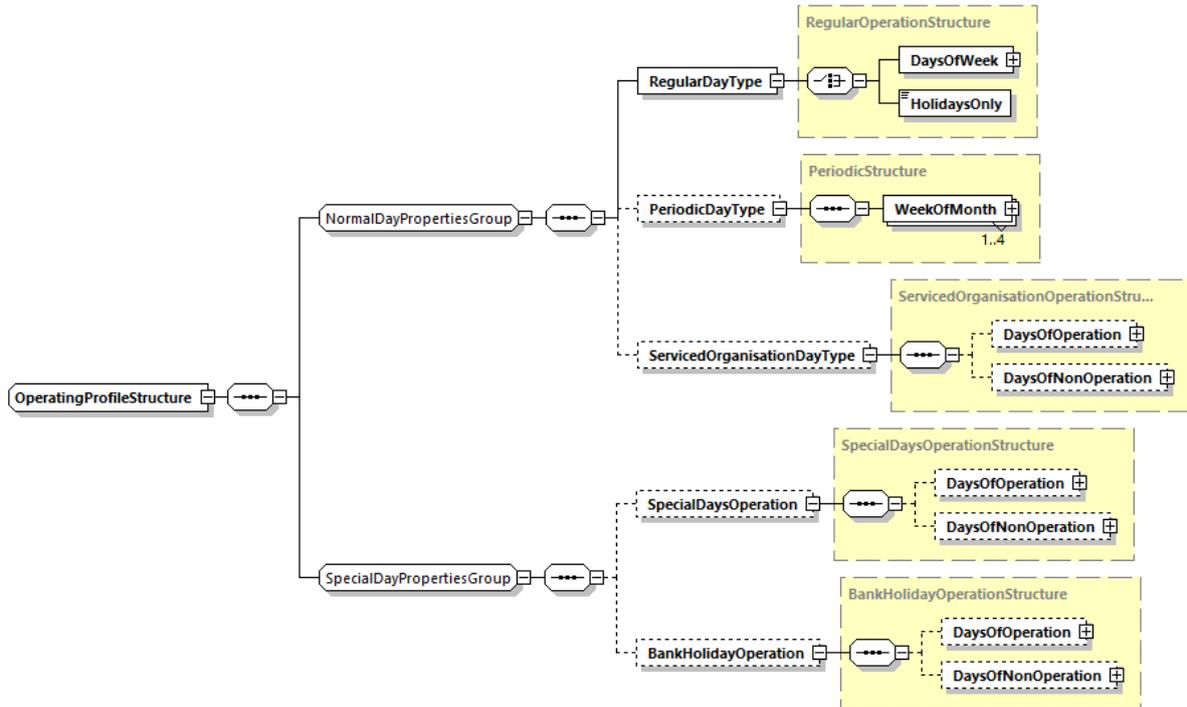*Figure 23 - Basic structure of an OperatingProfile*

As with TXC, in the event of conflict between days of operation and days of non-operation, the days of non-operation shall take precedence i.e. in the case of doubt, a trip shall be taken as *not* operating.

The usage of these within TXC-PTI is described in the sections below.
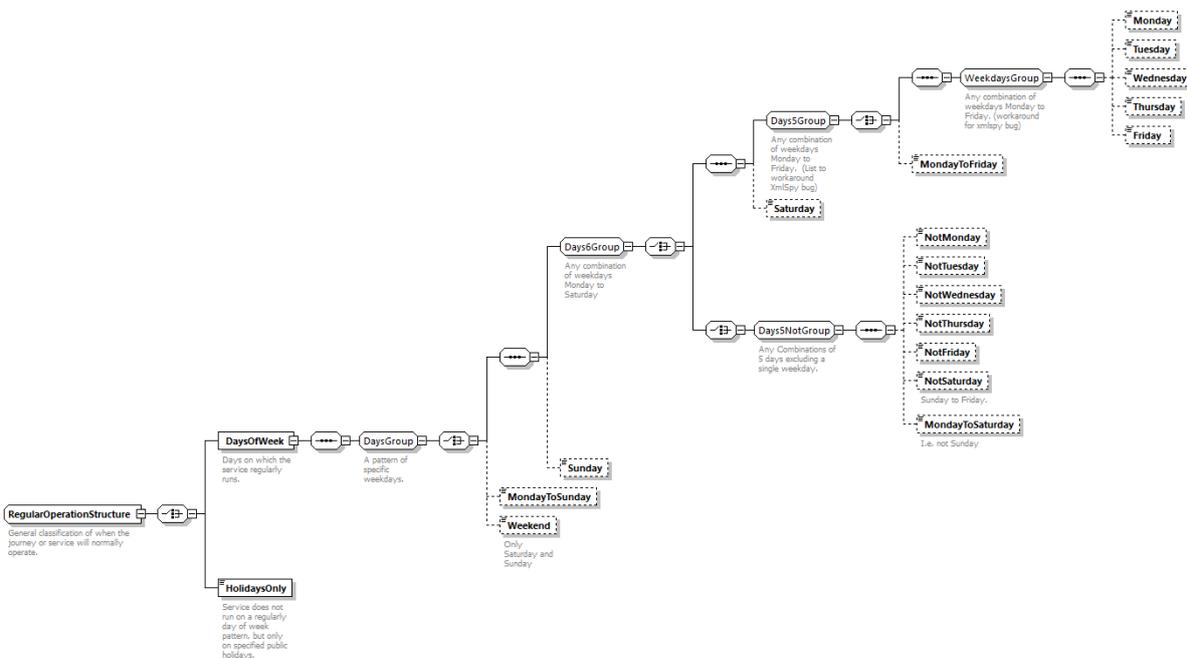
9.3.2   Regular days

Figure 24 - The RegularOperationStructure

In TXC, there is an expectation that some degree of regular operation will be defined. Indeed, it is mandated by the schema. There is a degree of confusion as to how the different elements all interact, however, and even more confusion when it comes to the groupings of regular days that can be used as enumerations.

Consider the structure for regular days shown in the figure above. Individual days are clear, as is the intent of some group enumerations such as *MondayToSunday*, *MondayToSaturday* and *MondayToFriday*. However, it becomes much less clear when looking at some of the other grouping enumerations. *NotSaturday*, for example, is annotated with text that describes it as "Sunday to Friday" but as part of a group identified as being a combination of 5 days excluding a single weekday, whereas "weekdays" is a group consisting only of the five individual days of the working week excluding Saturday and Sunday. Furthermore, *NotThursday* (for example), which is unannotated, would suggest a similar structure to *NotSaturday* i.e. Friday to Wednesday. This type of operation seems unlikely.

Because of this potential confusion, and the principle in TXC-PTI to make data clear and specific wherever possible, TXC-PTI shall therefore only use named, individual days of the week and shall not use the group enumerations.

The following example illustrates the use of **RegularDayType** to define a service which operates Mondays, Wednesdays and Fridays.

```
<RegularDayType>
  <!-- Not Tuesdays or Thursdays -->
  <DaysOfWeek>
    <Monday/>
    <Wednesday/>
    <Friday/>
  </DaysOfWeek>
</RegularDayType>
```

Note, particularly, that it is not correct in TXC-PTI to define all days of the week and then remove Tuesdays and Fridays through **SpecialDaysNonOperation**.

> In TXC-PTI, all regular days of operation shall be explicitly stated. Grouping enumerations shall not be used.

9.3.3   Periodic days

In TXC, periodic days are even less well understood, and consequently are rarely used. These define services which operate on e.g. 1st and 3rd Wednesdays of each month, and are often to be found in community transport services taking rural passengers to town market days. These are not their only uses, however.
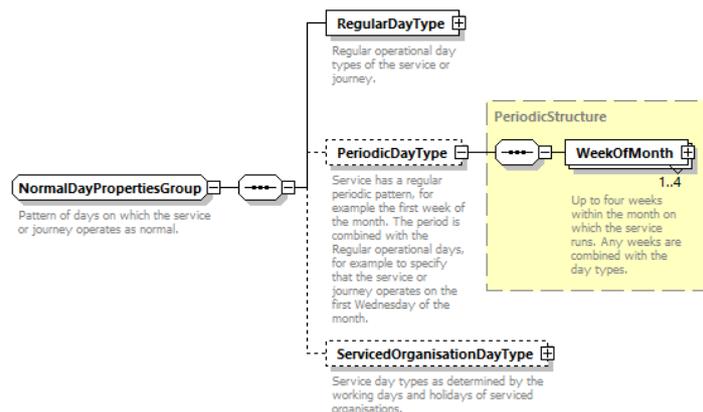
Figure 25 – The NormalDayPropertiesGroup structure, highlighting periodic days

Traditionally, TXC has tended to encode these services with specific **SpecialDaysOperation** dates. There are, however, two issues with this. The first is that simply having a set of dates does not provide any sort of context as to why those particular dates; the second, potentially more problematic for an operator maintaining data, is that the files have to be continuously refreshed as sets of date ranges expire.

For that reason, in TXC-PTI wherever a trip or service operates on regular, periodic days, then this shall be defined using the **PeriodicDayType** element available in TXC. This element works in conjunction with the **RegularDayType** element: **RegularDayType** states which day(s) of the week the trip or service operates, while **PeriodicDayType** indicates, through a set of week number enumerations, which weeks in that month it operates.

As with all operational day elements, serviced organisations and days of non-operation further qualify these dates – see relevant sections for details.

The previously quoted example, of a service which operates on the 1st and 3rd Wednesdays of each month, would be coded like this in TXC-PTI:

```
<RegularDayType>
   <!—Wednesdays only -->
   <DaysOfWeek>
      <Wednesday/>
   </DaysOfWeek>
</RegularDayType>
<PeriodicDayType>
   <!-- Repeated WeekOfMonth elements, one for each week of operation -->
   <!-- Here,first and third so that overall effect is 1st & 3rd Wednesdays -->
   <WeekOfMonth>
      <WeekNumber>first</WeekNumber>
   </WeekOfMonth>
   <WeekOfMonth>
      <WeekNumber>third</WeekNumber>
   </WeekOfMonth>
</PeriodicDayType>
```

Note that the **WeekNumber** element is an enumeration, and not a number. Possible values are *first*, *second*, *third*, *fourth*, *fifth*, and *last*.

In TXC-PTI, all regular periodic operation shall be coded using the **PeriodicDayType** element, and not with individual date ranges.

9.3.4   Special days

Special days are the method by which additional regular, irregular, or one-off dates or exclusions can be set up for trips. Such trips can include festival services, shuttle services which only operate when points of interest are open, football services, and similar.
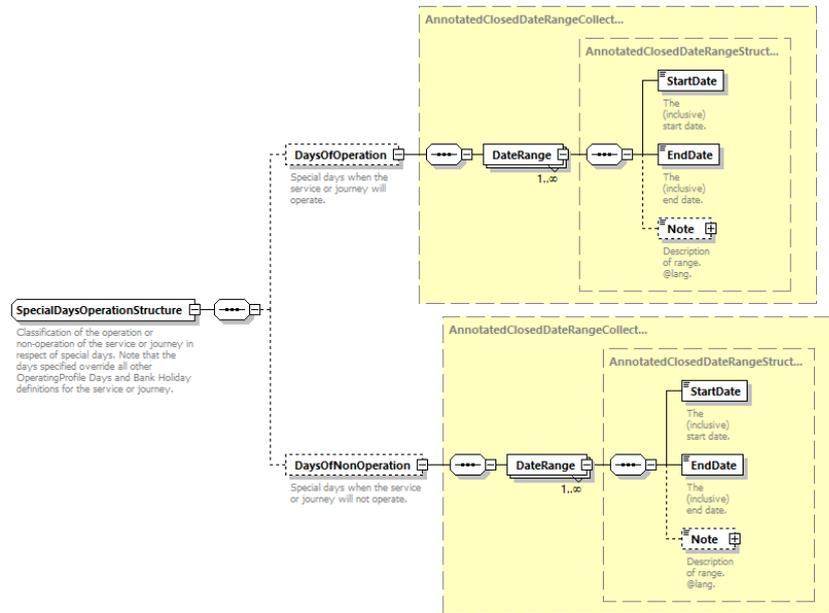

Figure 26 - The structure of special days

Special days are structured with two distinct sections: **DaysOfOperation**, describing the additional dates that the trip or service operates; and **DaysOfNonOperation**, describing the dates when the service does not operate. In both cases a series of date ranges is included, with inclusive **StartDate** and **EndDate** elements and a **Note** if required.

Note that, in TXC and consequently also in TXC-PTI, special days **DaysOfOperation** are *additive*; that is they operate in addition to any other stated days. This is true even if the special day occurs on a different day of the week to the stated regular days. This is particularly important to note if a range of dates is defined, since the trip or service would operate on *every single day within that date range*.

Where a trip or service is only operating on a particular set of dates, and does not have a regular day type, the coding **HolidaysOnly** shall be used in **RegularDayType** in place of **DaysOfWeek**.

For example, the following code would define a trip which operated only during a four-day event:

```
<OperatingProfile>
  <RegularDayType>
     <!—No regular operation -->
     <HolidaysOnly/>
  </RegularDayType>
  <SpecialDaysOperation>
     <DaysOfOperation>
        <!-- Festival operation – trip operates on all dates in range -->
        <DateRange>
           <StartDate>2020-07-30</StartDate>
           <EndDate>2020-08-02</EndDate>
        </DateRange>
     </DaysOfOperation>
  </BankHolidayOperation>
</OperatingProfile>
```

On the contrary, **DaysOfNonOperation** are *restrictive* (or *subtractive*); that is, they restrict the already stated days of operation. These could cover, for example, one-off events such as marathons which take place on known dates but which temporarily disrupt services. For example:

```xml
<OperatingProfile>
   <RegularDayType>
      <!-- Sunday operation -->
      <DaysOfWeek>
         <Sunday/>
      </DaysOfWeek>
   </RegularDayType>
   <SpecialDaysOperation>
      <DaysOfNonOperation>
         <!-- Great North Run 2020 -->
         <DateRange>
            <StartDate>2020-09-13</StartDate>
            <EndDate>2020-09-13</EndDate>
         </DateRange>
      </DaysOfNonOperation>
   </BankHolidayOperation>
</OperatingProfile>
```

The general principle in TXC-PTI is that dated ranges should only be used as a last resort. Often, trips are better coded, and contain more contextual information, using regular day types, periodic days, named bank holidays (whether using built-in enumerations or especially created) or serviced organisations. See other sections for more detail.

In TXC-PTI, special operating dates shall be used only as a last resort when all other options have been explored.

9.3.5   Bank holidays

Bank holiday enumerations have proved to be problematic to some users, particularly those in Scotland. This is because systems have tended to use groups of bank holidays, such as *AllBankHolidays* to simplify the output. In certain jurisdictions, though, these groups contain bank holidays with assigned dates that are not universally recognised. This causes normal operations to drop out of systems unexpectedly and results in significant work to correct.

One solution that was proposed is to dispense with the encoded bank holidays entirely and to rely on dated ranges to state when trips did and did not run. However, this is to throw out a significant degree of context provided by e.g. "does not run on Good Friday".

An alternative solution proposed using facilities in TXC to define only those bank holidays that were required, using special date ranges. However, this option was also discounted as it could lead to conflicts when named bank holidays were created for holidays that were already enumerated.

TXC-PTI has therefore adopted the principle that grouping elements shall not be used, and all bank holidays shall be explicitly stated in their appropriate **DaysOfOperation** and/or **DaysofNonOperation** structures. This is in line with the principles already discussed for regular days of operation above, and allows contextual bank holiday operation to be provided while removing problematic bank holidays from consideration.

Note that nothing in the foregoing prevents additional bank holidays being defined using the **OtherPublicHoliday** element, as some special days are introduced from time to time such as the Royal Wedding and Queen's Jubilee additional days in 2012. For the structure of the **OtherPublicHoliday** element, please refer to the TXC schema guidance.

For an example of bank holiday coding, consider the following snippet of XML. As structured, this would prevent a trip from operating Christmas Day, Boxing Day, New Year's Day, the 2ⁿᵈ January holiday in Scotland, and Easter Monday, but would allow it to continue running on St Andrews Day and a Scottish summer bank holiday, neither of which are routinely observed.

```xml
<BankHolidayOperation>
   <DaysOfNonOperation>
      <ChristmasDay/>
      <BoxingDay/>
      <NewYearsDay/>
      <Jan2ndScotland/>
      <EasterMonday/>
   </DaysOfNonOperation>
</BankHolidayOperation>
```

In TXC-PTI, all bank holidays with enumerations shall be explicitly stated within the **DaysOfOperation** and/or **DaysOfNonOperation** blocks. Grouping enumerations shall not be used.

Note that, in TXC and consequently also in TXC-PTI, bank holiday **DaysOfOperation** are *additive*; that is they operate in addition to any other stated days. This is true even if the bank holiday occurs on a different day of the week to the stated regular days. For example, the following code would define a trip which operated "Sundays and Bank Holidays (excluding Christmas and New Year)" in England:

```xml
<OperatingProfile>
   <RegularDayType>
      <!-- Sunday operation -->
      <DaysOfWeek>
         <Sunday/>
      </DaysOfWeek>
   </RegularDayType>
   <BankHolidayOperation>
      <DaysOfOperation>
         <!-- Additional bank holiday operation -->
         <GoodFriday/>
         <LateSummerBankHolidayNotScotland/>
         <MayDay/>
         <EasterMonday/>
         <SpringBank/>
      </DaysOfOperation>
   </BankHolidayOperation>
</OperatingProfile>
```

On the contrary, **DaysOfNonOperation** are *restrictive* (or *subtractive*); that is, they restrict the already stated days of operation. For example, a trip which is Mondays to Fridays excluding bank holidays would be coded shown on the following page.

```xml
<OperatingProfile>
  <RegularDayType>
    <!-- Weekday operation -->
    <DaysOfWeek>
      <Monday/>
      <Tuesday/>
      <Wednesday/>
      <Thursday/>
      <Friday/>
    </DaysOfWeek>
  </RegularDayType>
  <BankHolidayOperation>
    <DaysOfNonOperation>
      <!-- Bank holiday non-operation -->
      <ChristmasDay/>
      <BoxingDay/>
      <GoodFriday/>
      <NewYearsDay/>
      <LateSummerBankHolidayNotScotland/>
      <MayDay/>
      <EasterMonday/>
      <SpringBank/>
    </DaysOfNonOperation>
  </BankHolidayOperation>
</OperatingProfile>
```

9.3.6   Referencing serviced organisations

The coding and the application of serviced organisations has already been covered in section 2.3 above, and reference was made there to the way in which they are encoded in TXC-PTI **VehicleJourneys**.

To reiterate the fundamental point made in that earlier section, in TXC-PTI a serviced organisation shall only be provided with dates for working days, and all other dates shall be assumed to be holidays. This means that trips which operate on working (e.g. school) days only are coded as **DaysOfOperation/WorkingDays**, while trips which only operate during the holidays will be encoded as **DaysOfNonOperation/WorkingDays**.

An additional consideration for using serviced organisation dates within a TXC-PTI file is the manner in which the **ServicedOrganisation** interacts with the regular days. This is something which causes a degree of confusion in TXC, and it is important to note that serviced organisation dates *restrict* (are logically ANDed with) the days of the week provided in the **RegularDayType** element. So, for schools, for example, which by and large operate Mondays to Fridays, this pattern can be stated as the regular days of operation. The dates provided for the serviced organisation then list the range of dates during which those Mondays to Fridays apply.

Thus, there is no need for a **ServicedOrganisation** to define an entire half-term's worth of Monday to Friday dates. It is far simpler to define a single date range for the half term (e.g. 5 September – 18 October) and use the **RegularDayType** to limit this to Mondays to Fridays within that date range. This is much simpler to code and to interpret / debug.

This principle also carries forward into bank holidays. For the most part, bank holidays usually fall within school holidays. However, depending on the date of Easter and the Early May Holiday, some bank holidays can occur during term time. Given that dates of non-operation further limit the standard operation, it follows that there is no need to exclude bank holidays from the range of working days.

For example:

```xml
<ServicedOrganisation>
   <OrganisationCode>HPFLD</OrganisationCode>
   <Name>Harrop Fold School</Name>
   <WorkingDays>
      <DateRange>
         <StartDate>2020-02-24</StartDate>
         <EndDate>2020-04-03</EndDate>
         <Description>Spring Term 2020 - second half</Description>
      </DateRange>
      <DateRange>
         <!-- Date range omits 9 May bank holiday -->
         <StartDate>2020-04-20</StartDate>
         <EndDate>2020-05-22</EndDate>
         <Description>Summer Term 2020 - first half</Description>
      </DateRange>
      <DateRange>
         <StartDate>2020-06-01</StartDate>
         <EndDate>2020-07-16</EndDate>
         <Description>Summer Term 2020 - second half</Description>
      </DateRange>
   </WorkingDays>
</ServicedOrganisation>
...
<VehicleJourneys>
   <VehicleJourney SequenceNumber="1">
      <OperatingProfile>
         <RegularDayType>
            <!-- Weekday operation -->
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
            </DaysOfWeek>
         </RegularDayType>
         <ServicedOrganisationDayType>
            <DaysOfOperation>
               <!-- Weekdays restricted by the dates for organisation HPFLD -->
               <WorkingDays>
                  <ServicedOrganisationRef>HPFLD</ServicedOrganisationRef>
               </WorkingDays>
            </DaysOfOperation>
         </ServicedOrganisationDayType>
         <BankHolidayOperation>
            <DaysOfNonOperation>
               <!-- Bank holiday non-operation further excludes 9 May -->
               <!-- (MayDay in 2020) -->
               <ChristmasDay/>
               <BoxingDay/>
               <GoodFriday/>
               <NewYearsDay/>
               <LateSummerBankHolidayNotScotland/>
               <MayDay/>
               <EasterMonday/>
               <SpringBank/>
            </DaysOfNonOperation>
         </BankHolidayOperation>
      </OperatingProfile>
      ...
   </VehicleJourney>
   ...
</VehicleJourneys>
```

In TXC-PTI, serviced organisation dates restrict the regular day types, and are themselves further restricted by bank holiday non-operation. Thus, working date ranges for serviced organisations shall be stated simply.

9.4   Trip timings

As has previously been stated, the assumption in TXC-PTI is that information relating to a trip shall be fully defined within a trip. However, for reasons of compactness and readability it can be desirable, at times, to define some of this information at a higher level. TXC-PTI therefore allows this to happen, so long as any difference in a **VehicleJourney** is indicated by a complete replacement of the relevant blocks.
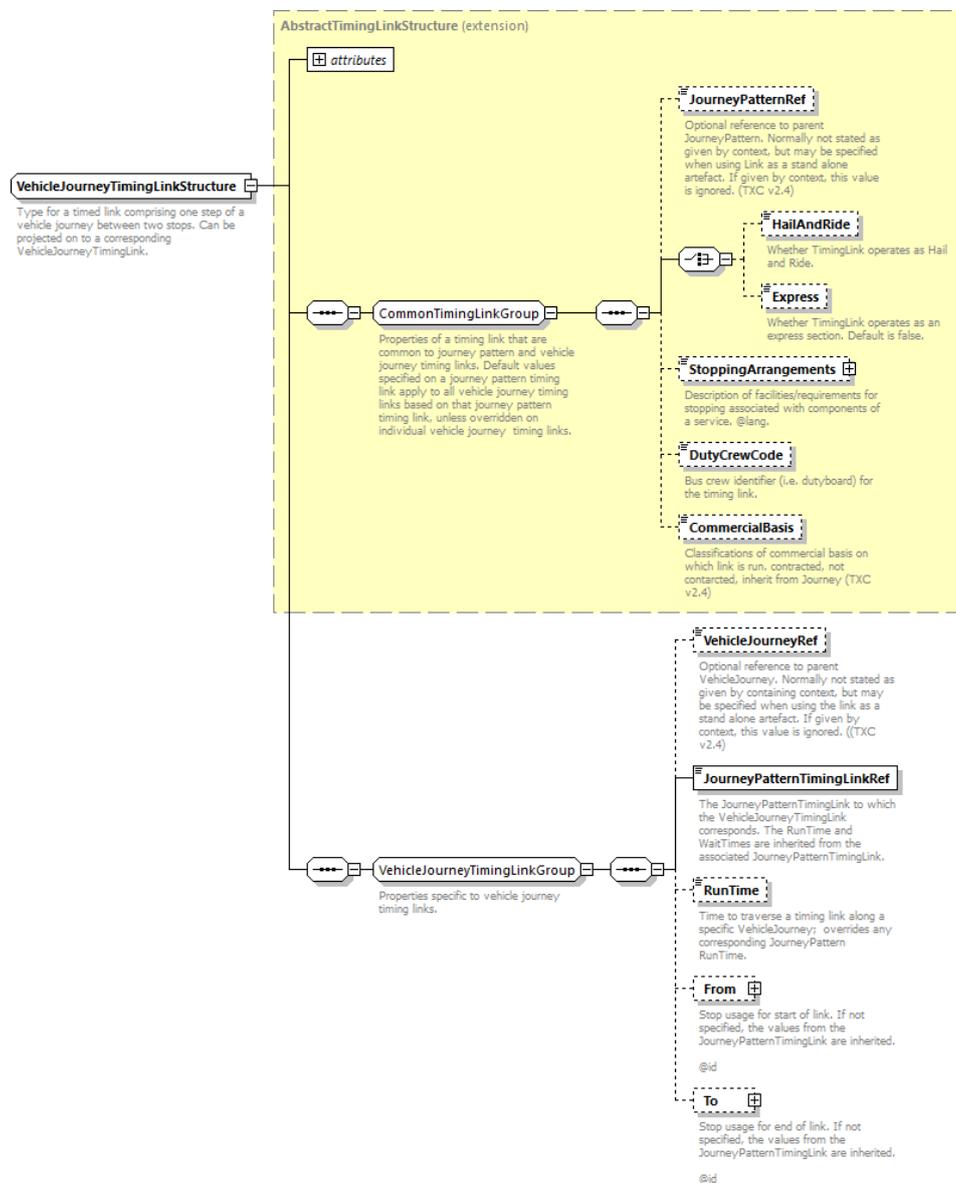

Figure 27 - The VehicleJourneyTimingLink structure

This is true for times at stop (or, more precisely, the times over the links in the vehicle journey). As described in section 8, a **JourneyPattern** can be defined with run times of zero minutes over each

link, with the actual times specified in each **VehicleJourney**. This is accomplished using a series of **VehicleJourneyTimingLink** elements, with the link times provided in the **RunTime** element, and the **WaitTime** (if any) defined in both the **To** stop at the end of one link and the **From** stop at the start of the next link.

A **VehicleJourneyTimingLink** has a similar – but not identical – set of elements a **JourneyPatternTimingLink**, and it is instructive to compare the structure shown above with Figure 20 in section 8.4. It can be seen that:

- A **VehicleJourneyTimingLink** has a reference back to the original **JourneyPatternTimingLink** via the **JourneyPatternTimingLinkRef** element.

- The **From** and **To** elements do not refer to the stop since this is inherited from the referenced journey pattern.

- In addition, for TXC-PTI the stop activity shall also be inherited from the journey pattern and not repeated in the vehicle journey. This is because, in general, it is unlikely that a stop will change whether it is set down or not (etc.) between trips. Note that timing point status is always inherited as there is no mechanism to override it in TXC.

- The common timing link group is common to both **VehicleJourneyTimingLink** and **JourneyPatternTimingLink**, and the same rules apply.

It should be noted that a **VehicleJourney** does not have the same Pattern > Section > Link structure as a **JourneyPattern**; **VehicleJourneyTimingLink** elements form a continuous sequence. For TXC-PTI, there can be no missing links in this sequence – there must be a one-to-one mapping between the links in a set of **VehicleJourneyTimingLink** element and the underlying **JourneyPattern** that is referenced by the **VehicleJourney**. If any links are to be omitted, then a corresponding **JourneyPattern** omitting the same links must exist.

> In TXC-PTI, where a default pattern is provided for timing links then this can be overridden by use of **VehicleJourneyTimingLink**. Where this is used then there shall be a one-to-one mapping to a **JourneyPattern** with the same set of links, and times shall be provided for all links and (where necessary) wait times. No links shall be omitted.

9.5   Journeys after midnight

TXC is a calendar-based system, and trips are defined as running on the actual calendar day on which they operate. However, many operators use an "operational day" concept, with days which start at (for example) 03:00 and run until 02:59 the following day.

For many services this leads to trips which run after midnight being shown at the end of a timetable panel, belonging to the previous calendar day. For example, trips in larger towns and cities, especially those with a sizeable student population, will often run trips in the early hours of Saturday morning and Sunday morning. These are shown, respectively, at the end of a Monday to Friday panel, labelled "Fridays only" and at the end of a Saturday panel.

These types of journeys are often wrongly coded in TXC, or are coded in a way that makes it difficult for a consuming system to place them in the right place when presenting the information in a traditional timetable. For example, TXC exists where:

- trips are shown with the incorrect day type.

In the example above, this would result in a trip that should operate in the early hours of Saturday morning being shown with a "Friday" operating date meaning that is being described, incorrectly, as operating in the early hours of Friday morning.

- trips are shown with the correct day type but cause unnecessary complexity on timetables.

  This is particularly true on a Monday, where trips on Sunday night into Monday are shown alongside similar, but differently timed, trips on other weekdays.

A number of different mechanisms were implemented in TXC v2.4 in order to overcome this limitation, some of which are more complicated than others. For example, there is scope for an operator to declare particular day types of their own and to map regular days and times into them. For TXC-PTI, though, this was considered too complicated and the aim was to have something that was simple to grasp, and not that different from the standardised, TXC v2.1, way of encoding data.

To that end, TXC-PTI trips which run after midnight shall use the **DepartureDayShift** coding. This is an offset that allows a trip to be coded to the correct operational day, but then given an offset as to when it actually runs. For TXC-PTI this shall only ever be "+1". Shifting departures backwards, or forwards by a greater amount, shall not be permitted.

For example:

```xml
<VehicleJourney>
   <!-- Trip runs operationally on Sunday -->
   <!-- Trip shall be displayed at the end of the Sunday timetable -->
   <!-- Trip physically runs at 00:15 on a Monday morning -->
   <OperatingProfile>
     <RegularDayType>
       <DaysOfWeek>
         <Sunday>
       </DaysOfWeek>
     </RegularDayType>
   </OperatingProfile>
   <VehicleJourneyCode>VJ1</VehicleJourneyCode>
   <ServiceRef>S1</ServiceRef>
   <LineRef>L1</LineRef>
   <JourneyPatternRef>JP1</JourneyPatternRef>
   <DepartureTime>00:15:00</DepartureTime>
   <DepartureDayShift>+1</DepartureDayShift>
</VehicleJourney>
```

and:

```xml
<VehicleJourney>
   <!-- Trip runs operationally on weekdays -->
   <!-- Trip shall be displayed at end of the Monday to Friday timetable -->
   <!-- Trip physically runs at 00:25 on Tuesday to Saturday mornings -->
   <OperatingProfile>
     <RegularDayType>
       <DaysOfWeek>
         <Monday/>
         <Tuesday/>
         <Wednesday/>
         <Thursday/>
         <Friday/>
       </DaysOfWeek>
     </RegularDayType>
   </OperatingProfile>
   <VehicleJourneyCode>VJ1</VehicleJourneyCode>
   <ServiceRef>S1</ServiceRef>
   <LineRef>L1</LineRef>
   <JourneyPatternRef>JP1</JourneyPatternRef>
   <DepartureTime>00:25:00</DepartureTime>
   <DepartureDayShift>+1</DepartureDayShift>
</VehicleJourney>
```

In TXC-PTI, operational days shall be described through use of **DepartureDayShift** to code trips to the correct operational and calendar days.

9.6 Vehicle Journey Interchanges

As discussed in section 5.3.6, interchanges are a key piece of information, and shall be included in TXC-PTI wherever possible. That section discusses how service-level interchanges are described, interchanging between journey patterns. However, this is not always possible – for example, not all trips connect - and there may be occasions when interchanges need to be described at vehicle journey level. This is achieved with a **VehicleJourneyInterchange**.

A **VehicleJourneyInterchange** is constructed in a very similar manner to the **JourneyPatternInterchange**, and it is instructive to compare the figure below to Figure 13 in section 5.3.6.

To start with the *CommonInterchangeGroup* is the same in both, and the **VehicleJourneyInterchange** elements within that shall therefore follow the same rules as set out for the **JourneyPatternInterchange** in section 5.3.6.2.

Beyond that, it is simply a case of specifying the incoming and outgoing vehicle journeys in place of the incoming and outgoing journey patterns, using **InboundVehicleJourneyRef** and **OutboundVehicleJourneyRef** elements, and specifying the stops at which the connection occurs using **InboundStopPointRef** and **OutboundStopPointRef**.

It is recommended that, in TXC-PTI, interchanges which can be made between trips on the same service, but which cannot be specified at service level, should be included using **VehicleJourneyInterchange** elements as described above.
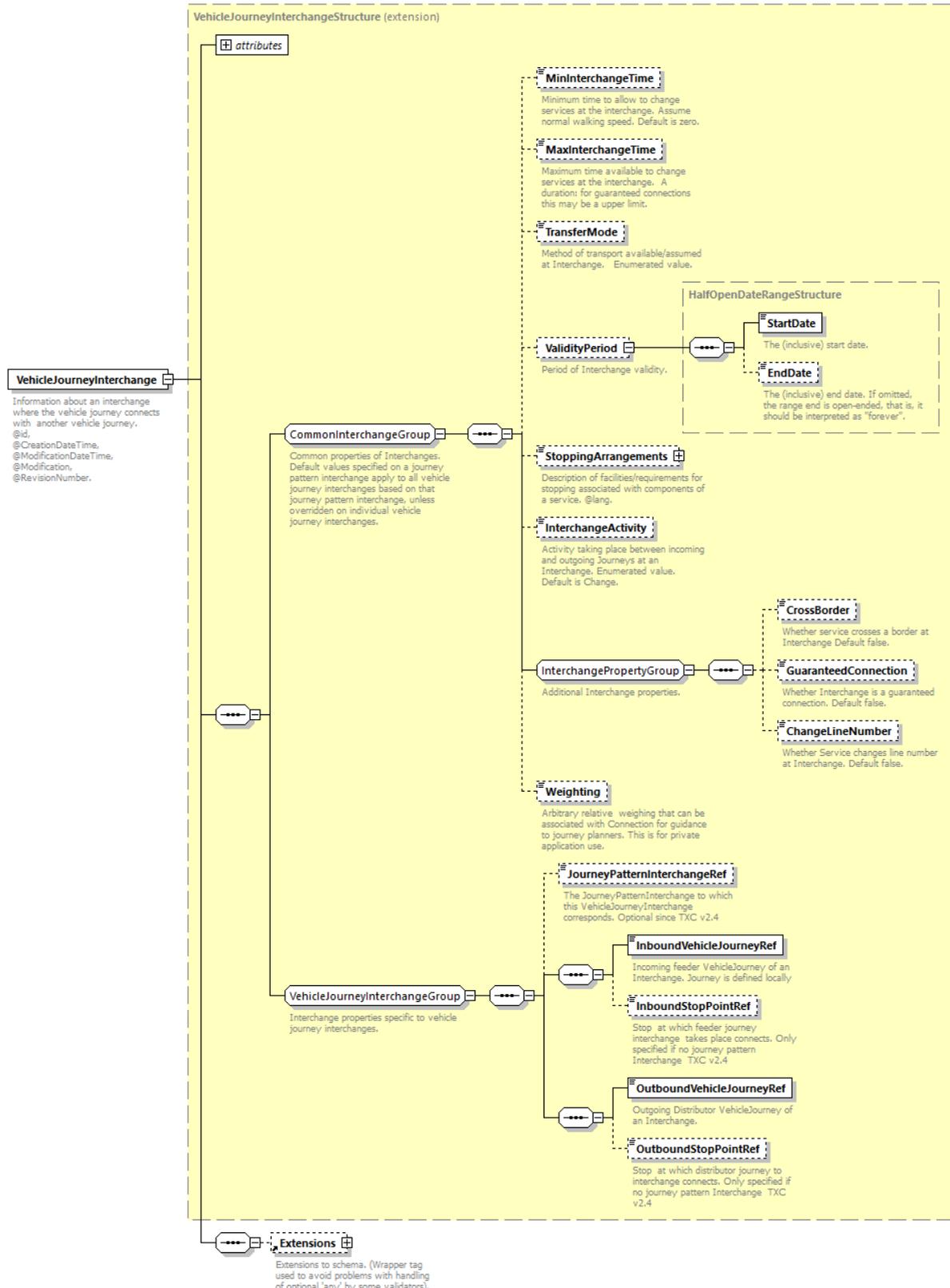
Figure 28 - The VehicleJourneyInterchange element

9.7   Real-time information

In a TXC-PTI document, information to support real time information shall be included where available. While this is not public-facing information, and TXC-PTI has been designed around information that is directly relevant to the public, these elements are nevertheless an important part of the information feeding real time systems.

Elements that may be populated in TXC-PTI include **GarageRef**, and the **Block** and **TicketMachine** elements within **Operational** element.

OperationalVariants, however, shall not be included. Where there are variants, these are best described using specific vehicle journeys.

In common with the principle of explicit declarations, where these are included, they shall be fully declared as part of a **VehicleJourney**, and not elsewhere.

There is merit in including dead runs in a **VehicleJourney**, as this helps real time systems estimate times for following trips. TXC provides both a **StartDeadRun** and an **EndDeadRun** for dead runs at the start and end of trips, and these are both allowable in TXC-PTI. Dead runs can take place from garage to the first stop for that vehicle, from the last stop for that vehicle to the garage, or between stops at the end of one trip and the start of another.
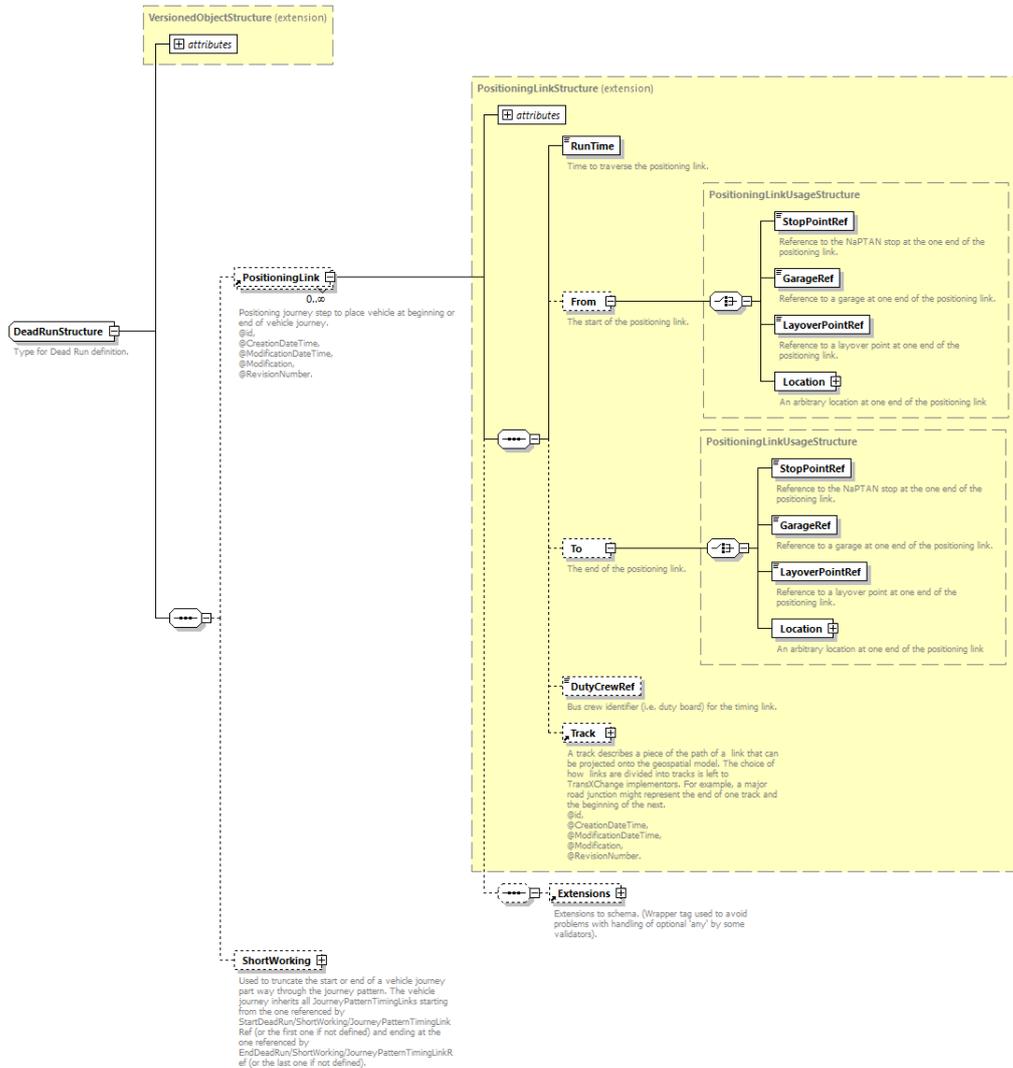

Figure 29 - The DeadRun structure

The structure of a dead run is shown in Figure 29 above. In TXC-PTI:

- there shall be a **PositioningLink**, which shall:
    - have a **RunTime** describing the run time between the **From** and **To** points;
    - have **From** and **To** elements, each of which shall be populated with either a **StopPointRef**, a **GarageRef**, or an arbitrary **Location**, although this last is discouraged in TXC-PTI unless absolutely necessary.
    - As with route links, the use of **Track** element to describe the actual path that the vehicle takes is strongly recommended, but not mandatory in TXC-PTI.
- **ShortWorking** shall not be used. Instead, a **JourneyPattern** which fully defines the short working shall be defined.

Given a specific time in TXC, and with a structure comprised of link run times, a system will calculate a call time at any given point by adding all of the run times and wait times together and using the result as an offset from the departure time. There is no expectation that a system will have to subtract anything. On this basis, for the avoidance of doubt, TXC-PTI defines the **DepartureTime** of a **VehicleJourney** as either:

a) the departure time from the first stop on the **VehicleJourney**, where there is no **StartDeadRun**; or

b) the departure time of the **StartDeadRun**, where there is a **StartDeadRun**.

> In TXC-PTI, information to support real-time systems shall be provided where available.

10 Flexible services

10.1 Declaring a flexible service

A **FlexibleService** is constructed, at a high level, in essentially the same way as a **StandardService**, and forms part of the **Service**, generally replacing the **StandardService**. It may be an addition to a standard service, but it seems unlikely that a non-flexible and a flexible service would be in the same file. In a **FlexibleService** there is a **FlexibleJourneyPattern** instead of a **JourneyPattern**. For this reason, a **FlexibleService** in TXC-PTI will have the same constraints as a **StandardService**, and a **FlexibleService** shall have at least one **FlexibleJourneyPattern**.

> In TXC-PTI, a flexible service shall have a **FlexibleService** which shall consist of at least one **FlexibleJourneyPattern**.
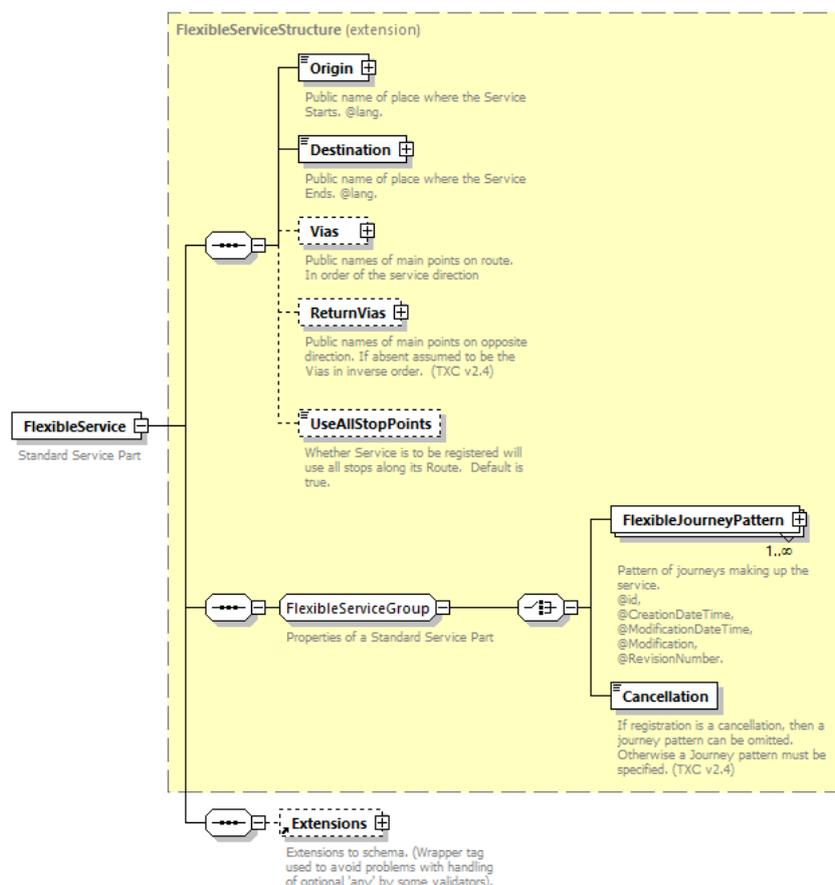
Figure 30 - The FlexibleService element

10.2 The FlexibleJourneyPattern element

At the top level, there is a high degree of commonality between a standard (non-flexible) service and a flexible service. The *CommonInterchangeGroup* is, in fact, common between both and a flexible service shall therefore follow the same rules for these elements – see section 8.2.

Beyond that, the flexible service is defined solely as an ordered set of stops, using the **StopPointsInSequence** element, and information shall be provided as to the booking arrangements needed to book the flexible service.

TXC-PTI does not need, at this time, to specify how these elements shall be completed. This will be the subject of a later revision.

In TXC-PTI, a **FlexibleJourneyPattern** shall consist of a set of stopping points, and shall provide information as to how the service shall be booked.

Figure 31 - The flexible journey pattern structure