NYSDOT Task 2.A Extension

Transit Schedule Data Exchange Architecture (TSDEA)

SDP Guidance Documentation
Part 2: User Requirements

Version 1.0

June 2008

By

ConSysTec Corp

Consensus Systems Technologies, Inc.

617-983-3364

Prepared for:
New York State Department of Transportation

Revision Table

Filename	Version	Author	Comments
Individual Chapters and Appendices	Draft 1.0	PEO	Draft Stakeholder Review
SDP_Guidance_Part2_v1.0	Final 1.0	PEO & LD	Preparing for Publication

Acknowledgements

This publication was prepared for the NYSDOT Statewide ITS Program, Task 2a Extension - Develop a Statewide, Standards-Based, Transit Schedule Data Exchange Architecture (TSDEA). It was developed under the Technical Support and Strategic Plan Development Services Contract of New York State. The *SDP Guidance Document - Part 2* is a compilation of the lessons learned and best practices based on the input of key downstate New York regional stakeholders. The Guidebook was made possible through the generous contributions of time and expertise by a diverse and large group of individuals representing their agencies. The individuals who composed the Regional Stakeholder Technical Working Group included:

*Andy Bata, NYCT
Arthur O'Connor, FHWA, US DOT
*Bruce Prusseck, Long Island Rail Road
Christine Falzone, CoachUSA
*Dave Rynerson, New Jersey Transit
David Glass, NYSDOT
*David Long, Metro-North Railroad,
David Markowitz, NYSDOT
David Rosenberg, NYSDOT
Don Galley, Telvent
*Douglas Benison, Long Island Bus
Ed Roberts, NYSDOT
Edward Mark, NYSDOT
Fredericka Cuenca, MTA Headquarters,
Gordie Peters, NYSDOT
Greg Bullock, NYCT,
Hetal Joshi, Telvent
Isaac Takyi, NYCT,
Jason Curate, NYCT
Judy Walker, MTA Headquarters,
Kuo-Ann Chiao, NYSDOT
Leokadia GLogowski, NYSDOT
Marc Boucher, NYSDOT
Marc Forgang, Telvent
Mary Marsh, NYSDOT
Michael Grattini, CoachUSA
Neil Trenk, Rockland County
*Paul Vollaro, formerly of MTA Headquarters,
Richard Mitchell, Long Island Bus
*Rob Bamford, Transcom
*Robert Campbell, Long Island Bus
*Robert Menhard, NYCT
Ron Epstein, NYSDOT
*Salvatore Mamone, NYSDOT
Sarah Kaufman, NYCT
Shelley Prettyman, MTA Headquarters,

*Steve Maus, Metro-North Railroad,
Steve Merritt, City of Poughkeepsie
Steve Riccardi, Long Island Bus
Tom Vaughan, NYSDOT
Vinny Pasquale, Westchester County / Beeline
Wendy Stubbs, Long Island Rail Road

Special thanks is extended to those individuals with a "" next to their names for going above and beyond participation in this effort.*

The Consensus Systems Technologies (ConSysTec) staff who wrote, incorporated comments and edited this Guidebook include:

Paula Okunieff
Nancy Neuerburg
Bruce Eisenhart

For her support and encouragement in the last phase of the project, the authors would like to thank Lisa Durst for her excellent and thorough editing and publishing efforts.

Finally, the ConSysTec staff would like to extend our thanks for the invaluable help, guidance and contributions of the NYSDOT Project Manager, Jim Davis, and his staff and colleagues: Tom Vaughan, Joy Chiu, Rich Quodomine and Matthew Sokol.

Table of Contents

List of Figures

List of Tables

Chapter 1: Overview of the SDP Requirements Manual

In This Chapter

- Learn about the scope of the SDP Guidance Documentation: Requirements Manual.
- Learn how to navigate through the Requirements Manual.

SDP Guidance Documentation: Requirements Manual

Audience: Analysts

Scope: Provide a detailed description of the framework and approach for the SDP, as well as a summary of the requirements that drove the development of the SDP.

Document Scope

This Schedule Data Profile (SDP) Guidance Documentation: Requirements Manual explains the transit schedule data requirements that drove the development of the Schedule Data Profile (SDP) XML Schema, and describes how native transit schedule data may be expressed using SDP data concepts and the SDP XML Schema.

This Requirements Manual is the second of four documents comprising the Transit Schedule Data Exchange Architecture (TSDEA) SDP Guidance Documentation. The first document is the Overview, which provides a high level overview of the SDP project, its background, goals and products. The Overview is intended for all readers. This Requirements Manual is targeted for analysts and assumes that the Overview has been read. The third section of the SDP Guidance Documentation is the Programmer's Manual, which provides additional technical guidance on the "how" of implementing the SDP. The fourth section is an interactive web site that allows peer-to-peer technical assistance.

General Topics in This Document

The contents of this Requirements Manual include 11 chapters and related appendices, which are listed below in four general topic areas or groupings:

Overview of the SDP Conceptual Data Reference Model and XML Schema

Chapters 2 and 3 present a high level overview of the SDP Conceptual Data Reference Model (CDRM) and the rules that were applied to represent it as a SDP XML Schema. Chapter 3 also describes how the CDRM may be used to represent the SDP data concepts as a physical database.

Detailed Description of Data Concepts

Chapters 4 through 11 discuss the detailed requirements of specific schedule-related transit data concepts. Appendix A focuses special issues encountered when working with rail or subway schedule data.

Topics on XML Schema

Chapter 11 discusses the rules and issues related to ensuring the conformance and validity of the SDP XML Document. Chapter 11 also discusses issues related to constraining or extending the SDP Schema so that it incorporates additional downstream requirements not anticipated in the original functional requirements document.

Appendices

Audience for SDP Requirements Manual

This SDP Guidance Documentation: Requirements Manual is targeted for program analysts. The discussion assumes that the reader has a basic understanding about how to read a data model (entity-relationship diagram) and XML Schema.[1]

How to Use This Document

This Guidance document incorporates the underlying requirements and business rules documented in the Functional Requirements document, including the Conceptual Data Reference Model (CDRM), and SDP XML Schema data concept, organization, and document requirements. The following is a guide for navigating through the chapters.

Chapter 2 provides a high level organizational view of the SDP Schema structure. It is a good place to start to learn how to navigate the SDP XML model.

[1] See Appendix C Resources for recommended tutorials on these topics.

Chapter 3 describes the Schedule Data Profile CDRM and the various ways that it may be implemented. The SDP XML Schema is only one way to implement the model, a physical database is another. This chapter defines the differences between the XML Schema and a typical physical relational database approach. The reader may find this chapter helpful in understanding the differences between the *abstract* (i.e., SDP CDRM) and *implementation* (i.e., XML Schema and physical relational database) approaches, as well as the differences among the implementation approaches. Although the chapter helps to describe the purpose of different modeling and implementation approaches, it is not needed to understand the rest of the document. In fact, it is very specialized for a technical reader.

Chapters 4 through 10 each describe a different set of related data concepts from the SDP CDRM. In particular, each chapter discusses the functional requirements from which the concept derives, the CDRM, and then the rules used to implement the abstract CDRM to the SDP XML Schema. Finally each chapter includes a section on how to apply native data to the SDP data concepts. Examples of SDP XML document fragments and business rules are discussed.

Chapter 11 discusses recommended conformance requirements and how to test conformance. It also describes how to create a "conformance contract," that is, how to extend or constrain the data described in the XML SDP document.

Appendix A identifies special issues encountered by rail data sets. The appendix may help rail professionals translate between transit bus centered language and its rail equivalent, when relevant.

Conventions and Types of Notation Used in This Document

This Requirements Manual uses several conventions and types of notations to describe the requirements and formats for the SDP data concepts.

- In an effort to maintain some consistency with prior SDP project documentation, many of the diagrams and tables are derived from the *Functional Requirements for the Schedule Data Profile* document.
- The Conceptual Data Reference Model uses the entity-relationship diagram notation.
- The SDP XML Schema uses a hierarchical tree-like structure based on the output of a commercial off the shelf tool (XMLSpy).
- Figure 2-4 in Chapter 2 provides a high level look how the XML Schema notation is used.
- The CDRM uses the entity-relationship diagram notation. See Appendix D for more detailed descriptions of the notations used.

Chapter 2: SDP XML Schema Model Overview

In This Chapter

- Learn about the organizational structure of the Schedule Data Profile.
- Learn the concepts included in each part of the SDP XML Schema Model.
- Identify the mandatory high level elements of the SDP XML Schema.
- Review SDP XML Schema notation.

Schedule Data Profile

The Schedule Data Profile (SDP) is a specification that describes operator generated schedule and related data. It is a business semantics specification that describes schedule information, specifically each data element and its relationship to scheduling data concepts, and preserves the referential integrity of these data concepts.

The SDP will be based on recognized information technology (IT) standards such as Extensible Markup Language (XML) and XML Schema, as well as standards and best practices in the IT and transit industries.

XML Schema Defined

An XML Schema is based on the e**X**tensible **M**arkup **L**anguage, XML, a general-purpose markup language.[2] There are many tutorials and descriptions of the language and schema standards that are helpful in understanding the SDP. A list of XML-related resources is included in Appendix C: Resources.

Purpose of the Schedule Data Profile (SDP) XML Schema

The SDP XML Schema is a structured description of the data concepts needed to describe key schedule-related data involved in transit service provision. The SDP XML Schema defines and constrains the format of an operating agency's schedule data and the relationships among the pieces of data. The industry-adopted XML and XML Schema standards, upon which the SDP was developed, provide a formal syntax that describes the "schedule" data requirements.

SDP XML Document

When schedule-related data is formatted into a file, following the rules of the SDP XML Schema, the file is called a SDP XML Document. This Guidance Document refers to a file of schedule data formatted using the SDP XML Schema as a "SDP Document" or "SDP XML Document."

Because formal XML standards have been adopted and implemented extensively, there are editing and validation tools readily available as off-the-shelf software that support native data extraction, loading, accessing, and storing (ELAS). In addition, many database management systems support these same ELAS functions.

Scope of the SDP XML Schema Model

The Schedule Data Profile consists of scheduling and related data that help describe a schedule. The scheduling elements are constrained to those elements needed for public information dissemination (e.g., timetable and trip planning functions), regional coordination and schedule planning activities. In particular, operator and track management functions are ***excluded*** from the current representation of regional schedule exchange. However, data concepts related to transit facilities and their location information are included in the scope of the SDP model.

[2] Link to resources with description of XML and XML Schema

Additionally, the model attempts to capture key metadata information on data quality, ownership, custodianship, lineage and currency. Metadata is discussed in greater detail in *Part 3, Programmer's Manual*.

Typology Used to Partition the SDP Conceptual Reference Data Model (CRDM)

Many different pieces of data and information are needed to accurately describe transit schedules and their relationship to service. In addition, some of the relationships between those pieces of data can be complex to describe. Partitioning a domain enables the organization of key ideas into smaller, less complex concepts. The SDP CRDM is partitioned to help users of the SDP better understand the data and data relationships. The partitioning of the SDP CRDM is based on the following categories, given that the data are organized by organization:

- Definition: General Agency Information
- Network: Transportation Network
- Place: Transit Feature Gazetteer and Transit Facilities
- Temporal: Service Provision

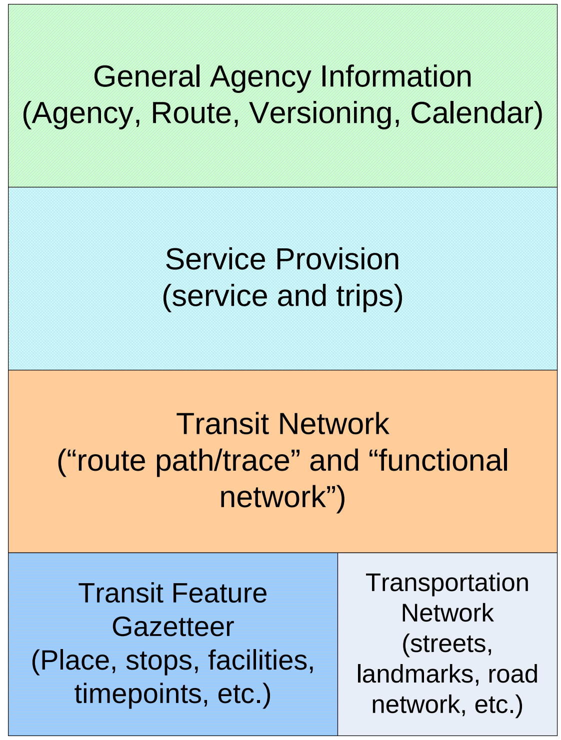
Places reference real-world locations, networks are made up of places, temporal services traverse the networks. Due to the complexity of transit "places," the "Place" category was divided into two—Facilities and Features (other than Facilities).

A fifth category (though not included in the SDP model) is the transportation network, a representation of the real-world network referenced by transit network and services. These categories are illustrated in Figure 2-1 below.

The partitioning helps distinguish different types of information the SDP must support. Building these separate blocks of information will eventually facilitate maintenance of the SDP model. Transit agencies update different types of information at different frequencies; additionally, various data sets require maintenance at different times. Partitioning the SDP Model into discrete logical categories identify the strong and weak relationships among the various scheduling elements. Elements within the same category are strongly related, and data in another category may be weakly related. Separating the elements also helps segment data that may be maintained at various times of the year. Stops may not be updated at the same cycle as service schedules. As such, building separate "branches" of information enables agencies to submit separate documents for different types of information.

Description of SDP Categories

The categories used to partition the SDP are further described below.



Legend
Light Green (right diagonal) = agency description
Light Turquoise (check) = service provision
Tan = transit network (transit features)
Pale Blue (horizontal stripes) = transit feature gazetteer
Grey = transportation network

Figure 2-1: Partition of SDP Domain

General Agency Information: This category describes general information related to the agency that registers schedule-related data and the actual data that are registered. Included in this category is information on the agency, schedule version and revisions, and the organization that submitted the data or is referred to by the data set content. The SDP XML Schema stores this information in the AgencyRegistration branch.

Service Provision: This category provides information on the service provided by an agency or organization unit of an agency. The scheduling elements include information on trips and trip times, scheduling notes, and bus assignment schedules. The SDP XML Schema stores this information in the Service branch.

Transit Network: This category describes the route path traversed by transit service. The network is composed of transit paths called patterns. Events occur on each pattern, and patterns may be composed of segments, often called route segments or timepoint intervals (TPI). The SDP XML Schema stores this information in the TransitNetwork branch.

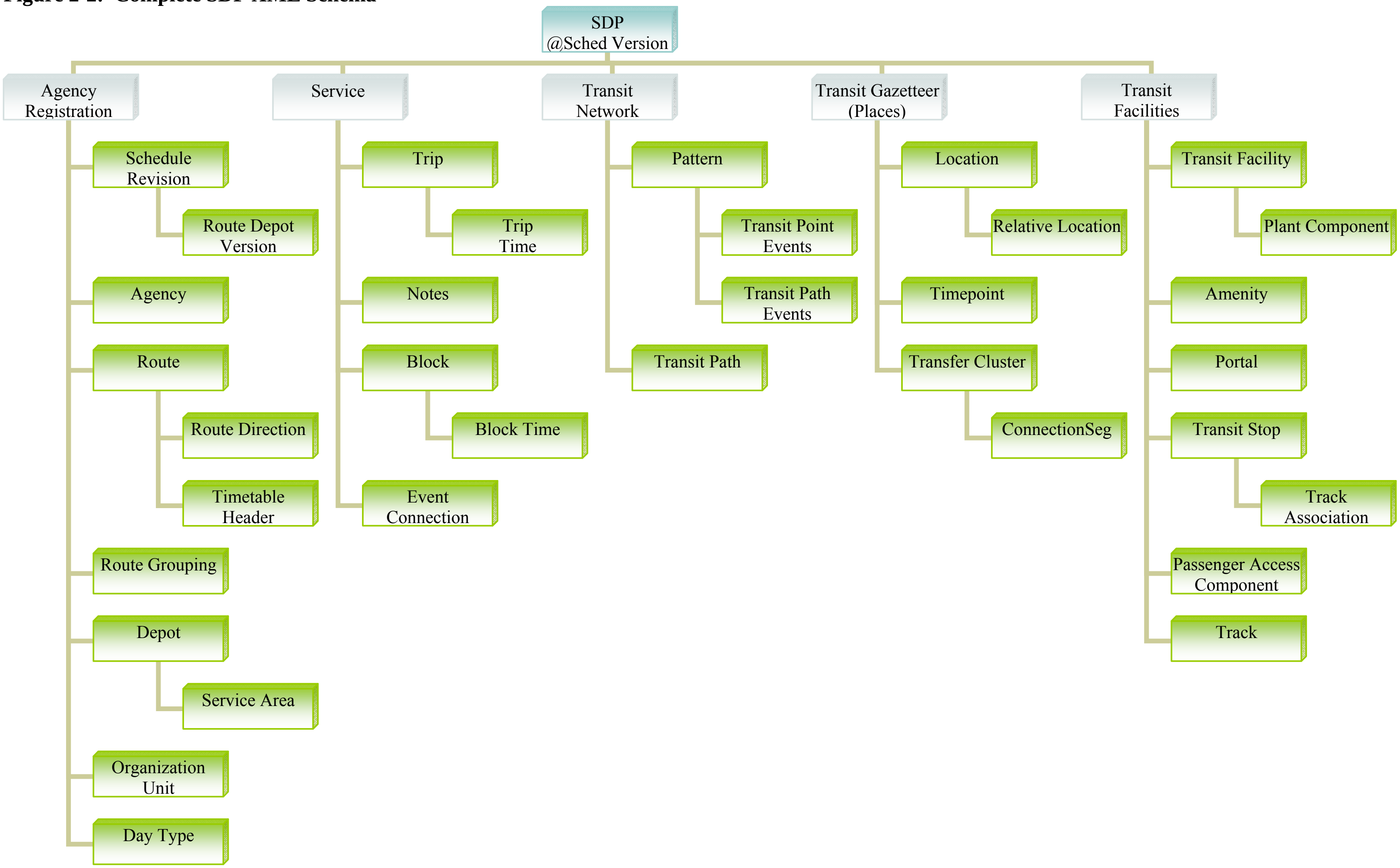
Transit Feature Gazetteer: This category defines places and their locations. A valuable addition is the incorporation of a "Location Table" which aggregates spatial references. The table simplifies data maintenance and enables the linking of places to equivalent locations (which may be described using a different location referencing system). For example, Penn Station Long Island Rail Road (LIRR) is the equivalent location as Penn Station Amtrak, New Jersey Transit, New York City Transit Lines A, C, E. Other transit features such as timepoints and transfer locations are also included in this category. The SDP XML Schema stores this information in the TransitGazetteer branch.

Transit Facilities: This category includes Transit Facilities and plant components that are included in or related to a transit facility. For example, parking lot, boarding area (platform), track, entrance, and elevator are plant components of a train station. The category includes the asset inventories related to transit facilities, that is, transit stops, amenities, portals, passenger access elements and tracks (associated with a transit facility). Due to the complex nature of key New York City transit facilities like Port Authority Bus Terminal, Jamaica Station, Grand Central Terminal, and Pennsylvania Station in New York, the model enables a facility to be part of another facility. The SDP XML Schema stores this information in the TransitFacilities branch.

SDP Categories Drive SDP XML Schema Organization

The partitioning of schedule-related data concepts drives the organization of the SDP XML Schema. The five areas related to the model partitions drove the categories used to separate the SDP schema into branches, accounting for all the scheduling elements described in the SDP Functional Requirements document. Each element in this SDP XML Schema model will be described in detail in subsequent chapters. As the abstract partitioning of the schedule-related data was transformed into an XML schema, some of the category names changed slightly. For example, Agency Registration is used in the schema while General Agency Information is used in the model above.

Figure 2-2: Complete SDP XML Schema



Mandatory SDP XML Schema Elements

The SDP XML Schema is designed to meet many existing and future needs. As a result, the Schema contains both mandatory and optional SDP XML schema elements. The SDP must meet the schedule information requirements of several downstream applications such as Trip Planning, Timetable, and Ad Hoc Schedule Planning[3], as well as support the existing upstream requirements of the data providers. Many data requirements were also included in the Conceptual Reference Data Model to allow extending the SDP to future downstream applications or to support current Operator practices. To support the diversity of requirements, there are many optional elements in the SDP Schema.

The high level mandatory (complex type) elements that are required by every validated SDP document are depicted in Figure 2-3. When implementing the SDP, the business rules described in each chapter must be carefully reviewed, because some additional elements from a different level of the SDP XML Schema may be required in the SDP XML Document.

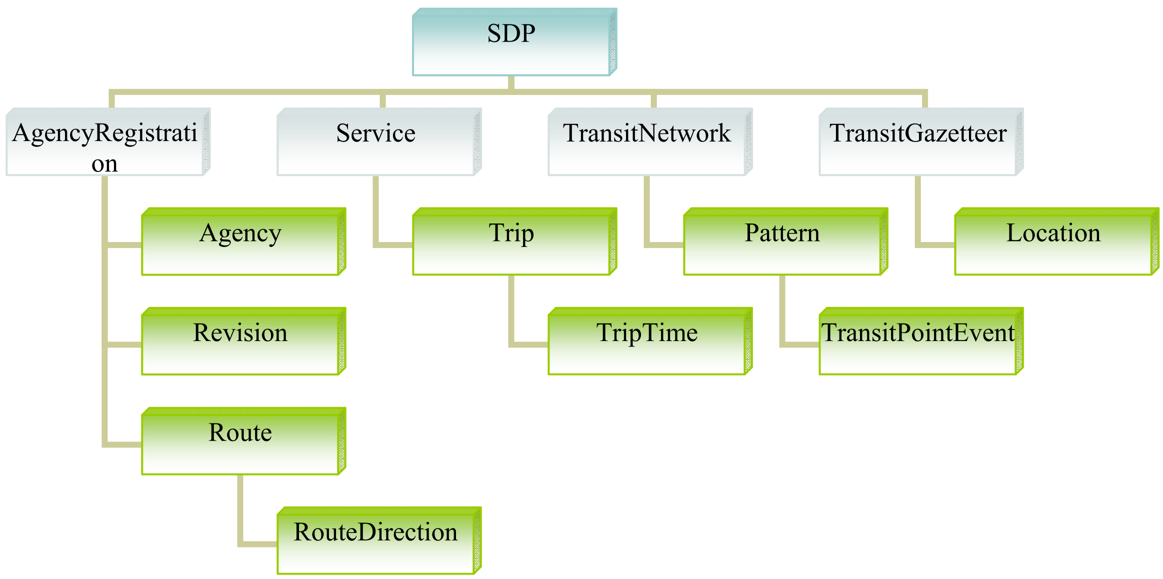


Figure 2-3: Mandatory SDP XML Schema Elements

XML Schema Notation

The XML Schema notation, as extracted from the XMLSpy application, is used to describe the organization and format of the SDP XML Schema. The Schema is based on a hierarchical organization where parent nodes or elements may contain child elements (which may in turn be a parent element to child elements). The XML Schema format and document instance are based on the standard notation of an XML Schema and instance document.

Figure 2-4 below illustrates the different levels of the XML Schema and key notation, using Transit Facility as the example. In addition, the figure shows the type description for each element. A type reference may have one of the following prefixes or suffixes:

- Prefix of "xsd" asserts the type is native to the XML standard

[3]The high level requirements for these applications may be found in the Appendices of the TSDEA Concept of Operations document.

- Suffix of "_id" implies the type is defined as an SDP identifier domain
- Suffix of "_cd" implies an enumerated code type.

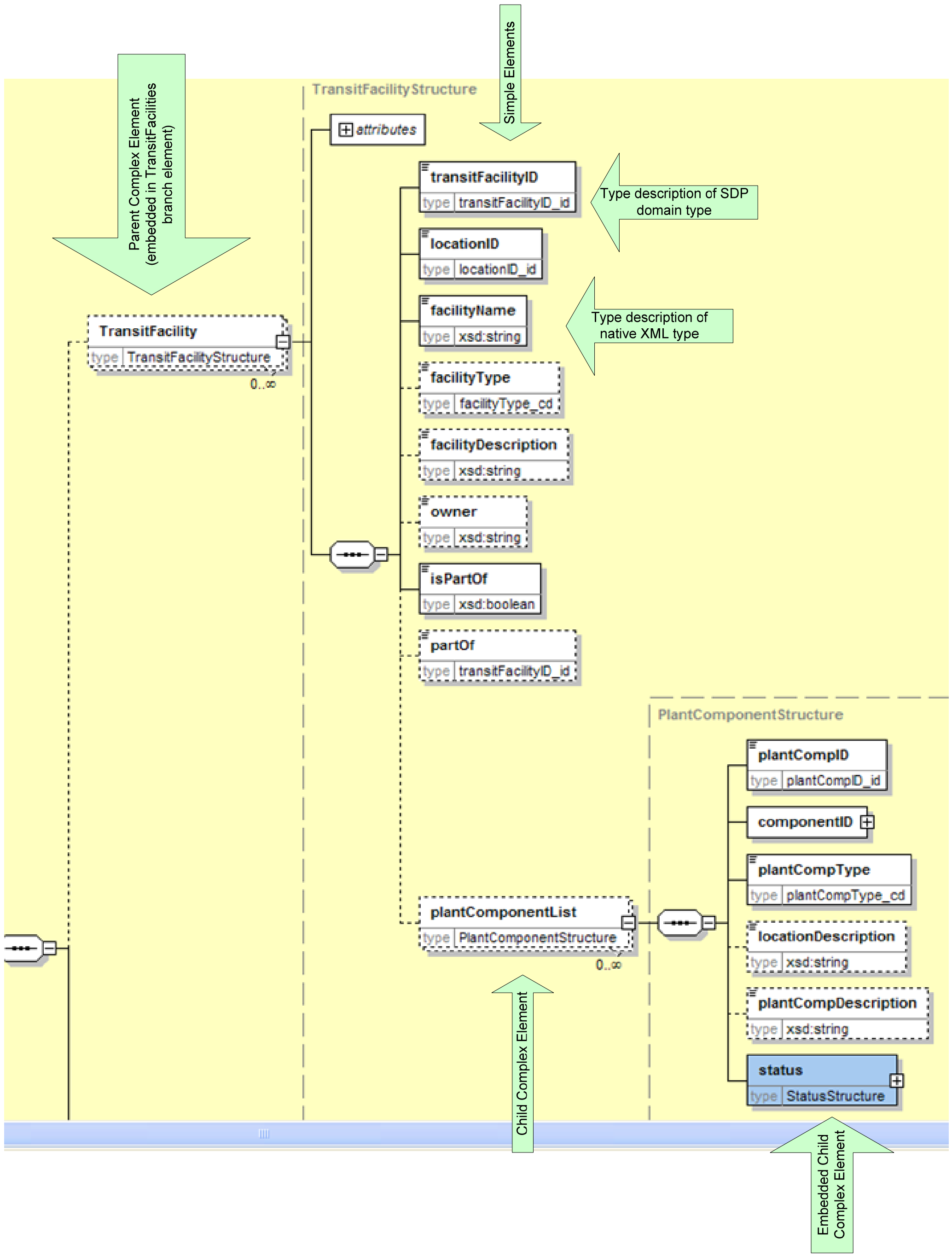


Figure 2-4: Example of the XMLSpy Diagram Notation

A "Structure" in the type name implies that the element is a complex type. An element also includes the constraint on the number of times it is allowed. An element enclosed by a dotted lined box indicates that the element is optional. Elements that may be repeated will include a notation of the minimum and maximum (e.g., 0..∞) under the right hand corner of the element enclosure. "plantComponentList" is an example of an element that is optional, but may be repeated. One element is required when the element is enclosed with a solid line (and does not contain a min-max value).

Chapter 3: SDP Conceptual Data Reference Model Overview

In This Chapter

- Learn about the purpose of a Conceptual Data Reference Model.
- Understand the methodology used to generate the SDP Conceptual Data Reference Model.
- Discover the business rules used to implement the Conceptual Data Reference Model as a physical or XML Schema representation.
- Learn about the differences among conceptual, logical, physical (entity-relationship) and XML Schema models.

Introduction to This Chapter

Although the two basic formats that are needed to understand the Guidance document are the SDP CDRM and the SDP XML Schema, there are other methods that may be used to implement the SDP Conceptual Data Reference Model. This chapter describes the purpose of the CDRM with respect to its *reference* representation as well as the methods used to implement it.

Schedule Data Profile

The Schedule Data Profile (SDP) is a specification that describes operator generated schedule and related data. It is a business semantics specification that describes schedule information, specifically each data element and its relationship to scheduling data concepts, and preserves the referential integrity of these data concepts.

The SDP will be based on recognized information technology (IT) standards such as Extensible Markup Language (XML) and XML Schema, as well as standards and best practices in the IT and transit industries.

Purpose of a Conceptual Data Reference Model

The purpose of a Conceptual Data Reference Model (CDRM) is to describe the "real-world domain" using unambiguously defined set of data concepts and model their relationship to each other. The technical language and methodologies used in the development of a CDRM can be confusing, in part, because of different uses of the same terminology.

Basically, the SDP's CDRM is an abstract model that describes the real world domain of transit scheduling. The technical approach or methodology used included entities, relationships and attributes, which are captured in a model that is independent of technology implementation. In other words, the CDRM can guide more than one way to implement the sharing of transit schedule data. For example, a database management system and SDP XML document are two methods that may be used to implement the SDP CDRM.

The effort may be likened to building a house in the Tudor style. We know we want a Tudor style house, that is, "steeply pitched roofs, half-timbering often infilled with herringbone brickwork, tall mullioned windows, high chimneys, jettied (overhanging) first floors above pillared porches, dormer windows supported by consoles."[4] We also need to develop a framework that supports both the style and the types of features needed by the family that will live there (number of rooms, heating system, etc). The prospective house may be modeled in a number of different ways, such as in a three-dimensional form, or a series of diagrams from multiple perspectives. Finally, we need to develop a set of blueprints and identify the specific

[4] From http://en.wikipedia.org/wiki/Tudorbethan_architecture.

materials that will be used to build the house. One set of materials may necessitate changes to the blueprint or impact other types of materials specified to be used to build the house.

The different information technology modeling approaches used to develop the SDP are no different. The SDP must be described in several different ways in order to support the fundamental needs and goals of the Schedule Data Profile. In this section, we describe the different modeling technologies and their role with respect to the final product, that is, the Schedule Data Profile XML document of schedule data submitted by each Agency.

The CDRM was developed to help ensure that the needs of the applications that use schedule data are met by the SDP. It identified key requirements related to the data validity including: identity, uniqueness, references, attributes, and data format and type. These requirements were captured in the SDP Functional Requirements and modeled in the SDP CDRM. As a result, the model helps ensures that data, when implemented and subsequently exchanged, are consistent and well understood across applications.

The SDP's CDRM accomplishes the following:

- Provides a reference for the meaning and relationships of the real-world domain concepts when they are implemented in applications and interfaces;
- Defines the identifiers and uniqueness requirements required for downstream applications;
- Provides a description of core attributes that are needed by most downstream applications;
- Incorporates flexibility to constrain or extend the model given various implementation approaches and tools.

Methodology for SDP Reference Model Development

The SDP Project used a system engineering approach for developing user driven requirements. The initial stage of the process involved soliciting stakeholder input on how key user groups currently use and might in the future use schedule and related information. This *Concept of Operations* (ConOps) was developed through stakeholder meetings and interviews. A set of downstream application operational descriptions, high level requirements, and detailed data requirements were documented in a series of white papers. The white papers (also called Use Cases) included downstream applications such as Integrated Trip Planning, Dynamic Generation and Presentation of Public Timetables, and Generation of Ad Hoc Scheduling.

The High Level requirements from the Use Cases provided the initial input into the scope and requirements for the conceptual reference model. A draft CDRM was developed, and through a series of stakeholder meetings, the model was refined in order to better capture user upstream practices and downstream application needs. The product from this effort was a comprehensive Functional Requirements document. The requirements document not only covered the schedule and related data concepts found in the reference model, but it also described a preliminary set of requirements for naming and formatting an exchange standard based on XML Schema standard, as well as integration issues for a central repository that fuses individual agency data into a regionally related data set.

The CDRM is meant to be used as a framework to unambiguously describe the SDP data concepts and their relationship to each other. Different technical methods may be used to physically represent and store schedule data. The three methods include: logical data model, physical database, and XML Schema. The first two models (logical and physical) are shown using an entity-relationship notation. Appendix D describes the symbols and conventions of the notation in more detail.

Differences Between the SDP CDRM and Implementation Methods

The SDP CDRM uses an entity-relationship (ER) method to represent real-world phenomena. The model is driven by a set of requirements described by current, local practice and by best practices advocated by the information technology and transit industries. The CDRM uses an abstract ERD modeling method to represent these real-world phenomena. Although a similar notation is used to describe the Logical and Physical models, they do not include the same information or serve the same purpose. Differences between the CDRM and Logical, CDRM and Physical and CDRM and XML Schema models are described below.

Difference between a CDRM and SDP Logical Entity Relationship Model: A CDRM shows the relationship between entities, but does not carry related keys to related entities. For example, in a system that supports more than one schedule version per agency, the schedule version identifier must be included in every entity in the logical model. A *logical model (expressed as an ER diagram (ERD))* shows these primary and foreign keys, and thus describes key storage requirements related to the data set. The CDRM makes no assumptions about how a model is applied, rather, it describes real-world relationships.

Difference between a CDRM and SDP Physical Implementation: Similar to the relationship between a conceptual and logical model, the physical implementation supports primary and foreign keys, the procedures that validate these relationships, and specific formats and syntax related to each data type described in the model. Specifically, these rules and procedures are defined for a specific database management system such as Oracle 9i, MS Access 2003, etc.

Difference between a CDRM and SDP XML Schema Implementation: The SDP XML Schema's primary purpose is to facilitate the sharing of data across different information systems, particularly via the Internet. The SDP XML Schema uses the CDRM to describe schedule and related data concepts and preserve the relationship requirements among data concepts for one schedule version and for a single transit operator. A set of rules were used to migrate the data concepts from the CDRM to the XML Schema implementation. The general set of rules are described below, detailed rules pertaining to each entity are documented in the following chapters.

Rules for Implementing the SDP XML Schema from the CDRM

The specific rules for migrating the CDRM to the SDP XML Schema implementation include the following:

- Each entity became a complex type. Some entities that define types of things, like day type, pattern type became an enumerated type or code embedded in the complex type.
- An entity that is related to another entity and typically acquires a foreign key in a logical model, may be dealt with in one of two ways.

  - The entity that is related to another entity is embedded as a child element in another element. For example, RouteDirection is embedded as element <routeDirectionList> in the RouteStructure.
  - The complexType description for the entity carries an element similar to a foreign key. For example, Trip includes elements routeID and patternID.
- If a related entity becomes a stand-alone element and carries a related key (like Trip includes patternID and routeID), then the element value is constrained using the "keyref" notation (as described in the XML Schema standard).
- Even if there is reason to embed an element into a parent, the XML Schema should not include more than four layers, that implies that the maximum number of children is less than four elements from the root node.

An Example of Migrating the CDRM to a Logical, Physical and XML Schema Representation

As described above, there are rules for implementing the CDRM from the conceptual framework to its logical, physical and XML schema formats. The following sections show how the CDRM for the same data concept, Schedule Calendar Date, is transformed to a logical, physical, and XML Schema model. Each model depicted in Figures 3-1 through 3-7, is summarized in the list below:

- Conceptual Data Reference Model—*Figure 3-1*
  - Note the CDRM, Logical and Physical models are all represented using ERD notation.
- Logical ERD Model—*Figures 3-2 and 3-3*
  - Note the key identifiers become primary keys (pk), and related entities include related or foreign keys (fk);
- Physical Model—*Figure 3-4*
  - Note the attributes are specified with specific data types that reference the specific database management system specifications;
- XML Schema—*Figures 3-5 to 3-7*
  - Note the data are formatted in a hierarchical format, and relationships must be maintained by internal functions.

Example of the Conceptual Data Reference Model

The CDRM is expressed as an Entity Relationship Diagram (ERD). Figure 3-1 shows an example for the basic representation of the schedule calendar date concept.

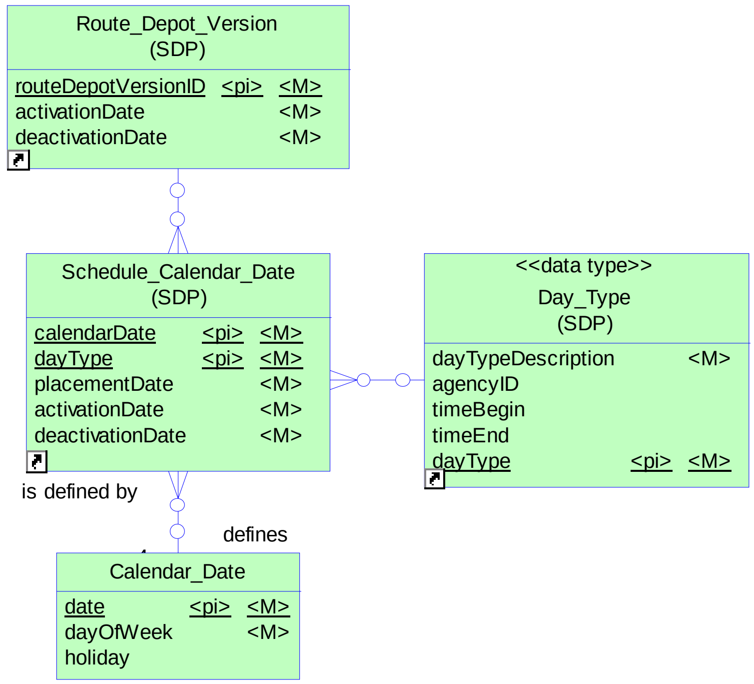


Figure 3-1: Conceptual ER Model of Schedule Calendar Date Concept

The following paragraph describes the requirements of Figure 3-1:

> "Transit service is scheduled for each day of operation. Service components may be scheduled to operate on different dates depending on a number of factors. These factors may be schedule based; for example, special trips are designated when there is an event at Shea Stadium or service to evacuate workers from the city during a snow storm. The Schedule Calendar Date associates the relevant schedule components (designated by the Route Depot Version) and an index related to the appropriate trips (designated by the day type) into a table which is used as a reference.
>
> "A Schedule Calendar Date is created for each set of schedule version components and the trips that operate on the specific dayType. In some cases, the schedule version components are scheduled for only part of a day, for example, the schedule components vary when the Mets play games that begin at 5 p.m. versus at 7 p.m." [*SDP Functional Requirements*, p. 102]

Example of the Logical Entity-Relationship Representation

The logical model is driven by application requirements related to how the data are stored and accessed. In the example illustrated in Figure 3-2, the Schedule Calendar Date entity inherits related keys designating the schedule version when more than one schedule version is present. When an organization changes its schedule mid-version, the entity is required to include a revision number, and when a transit agency issues their schedule by route or by route and depot, the route-depot version is also included in the entity. When this model is extended to a regional repository, each entity must designate the authority that issued the data, as such, the functional entities Route_Depot_Version, Schedule_Calendar_Date and Day_Type include the agency identifier (agencyID). The actual implementation of the conceptual to logical model may be seen in Figure 3-6.

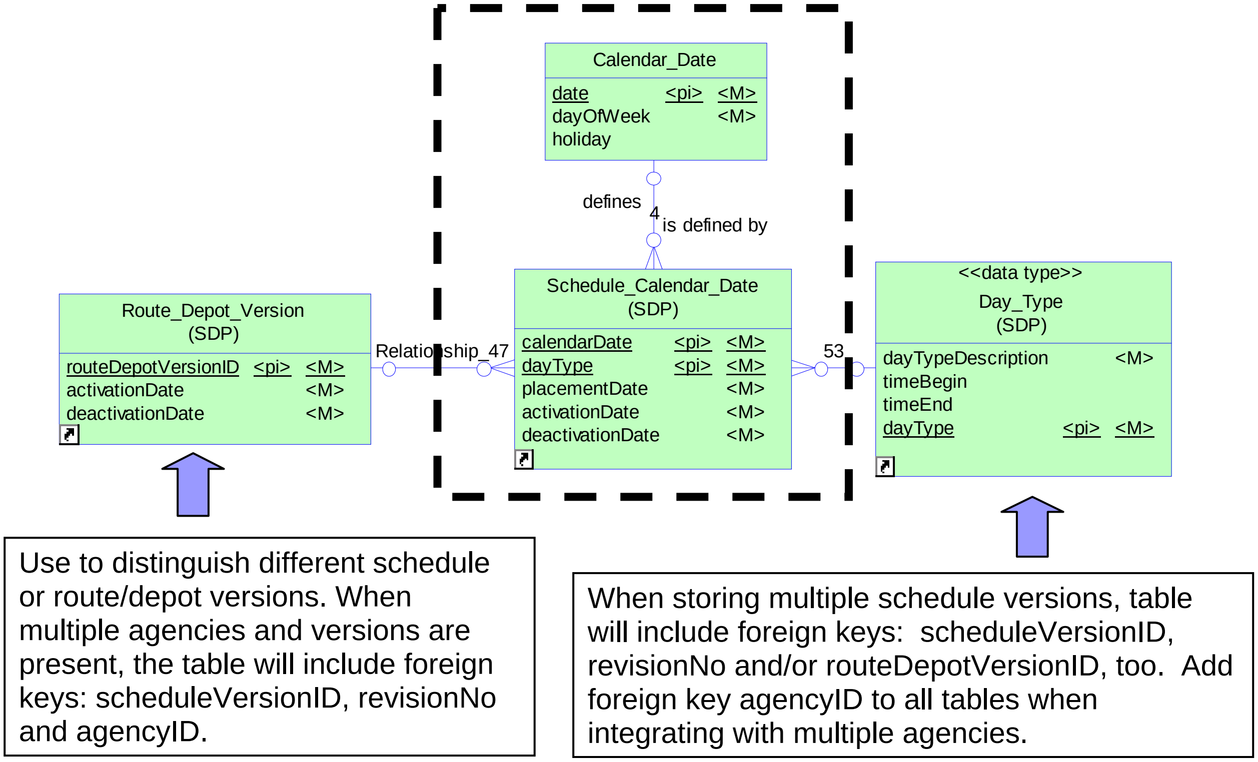


Figure 3-2: Migrating From Conceptual to Logical Model

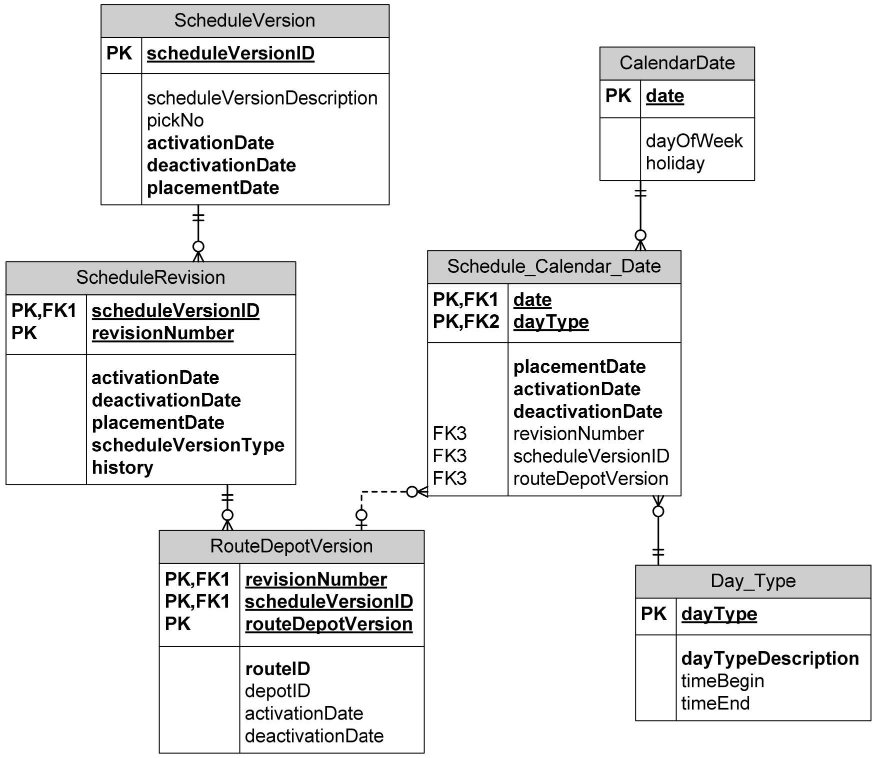


Figure 3-3: Logical Model of Schedule Calendar Date Concept

Example of the Physical Database Implementation

The physical model is similar to the logical model except the data types are defined by the database management system. The physical database also supports procedures that enforce referential integrity triggers (primary and foreign keys) when data are added, changed or deleted from the database. A generic physical model for the Schedule Calendar Date concept is illustrated in Figure 3-4. As is shown in the figure, this is similar to the logical model shown in Figure 3-3 **except** for defining specific data types and showing the procedures. For organizations with specific database management systems, the logical and physical representations are somewhat redundant since a logical and physical models will use the same data type definitions.

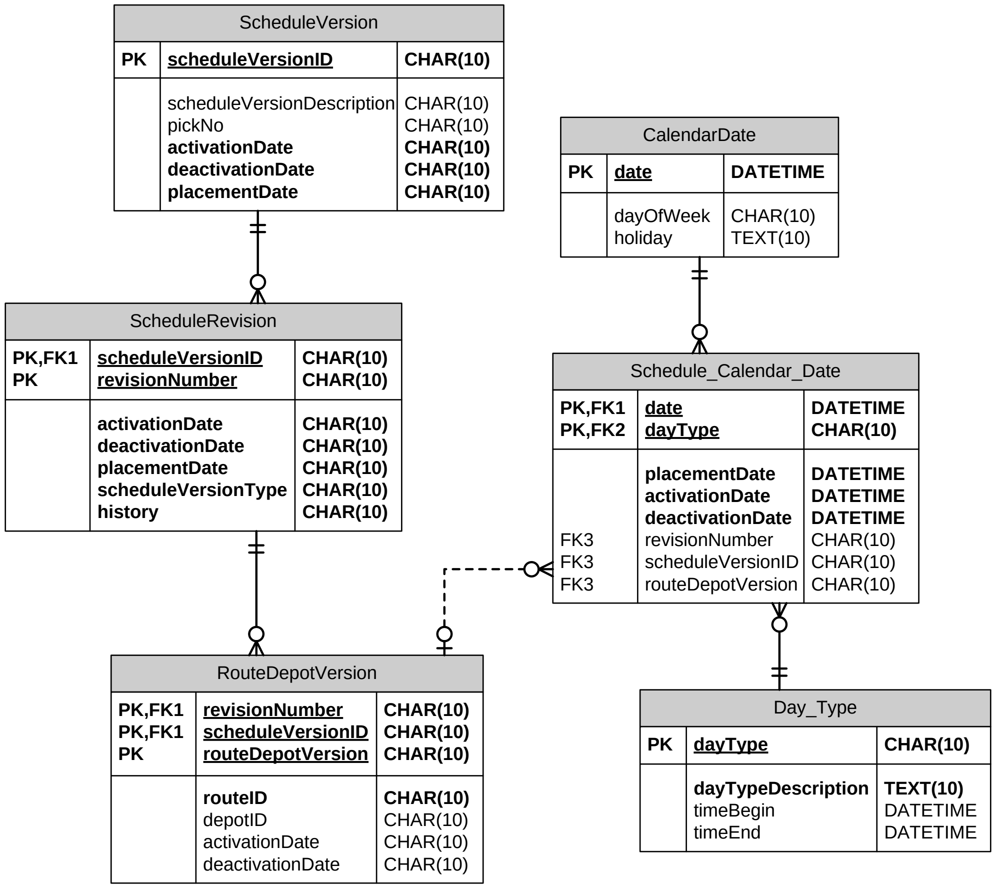


Figure 3-4: Physical Model of the Schedule Calendar Date

Example of the XML Schema Implementation

An XML Schema is organized like a hierarchical database. In the case of the special Schedule Calendar Date, the schema reflects two key elements (since the schedule version, revision and route depot version elements are defined in the SDP schema)[5]. The two elements are depicted in Figure 3-5.

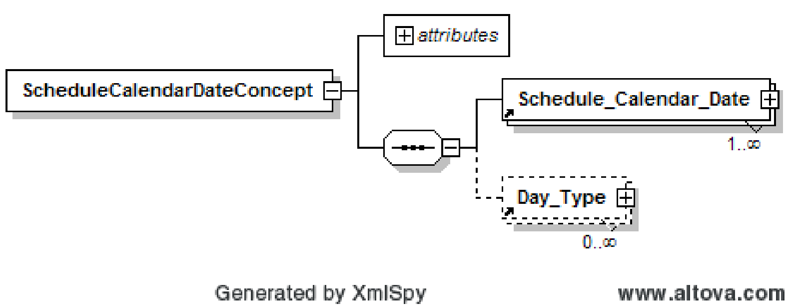


Figure 3-5: Schedule Calendar Date Concept XML Schema

[5] This example does not reflect the current Schedule Calendar Date (SCD) Schema. The DayType element is now included in the SDP XML Schema under the AgencyRegistration branch and is no longer included in the SCD Schema.

Figures 3-6 and 3-7 show the details of each element, Day_Type and Schedule_Calendendar_Date, respectively.

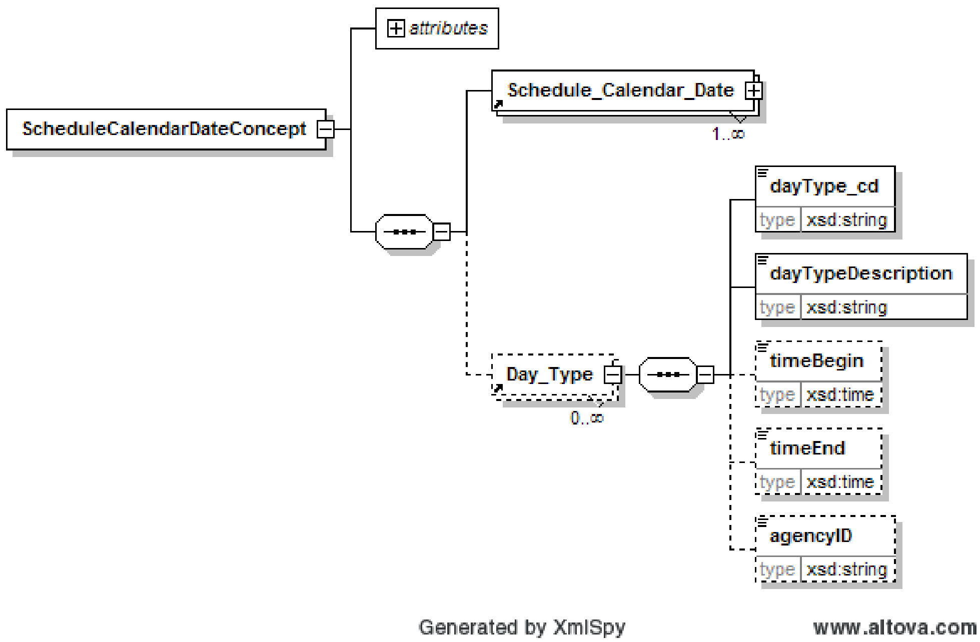


Figure 3-6: Day_Type representation in the Schedule Calendar Date Concept XML Schema

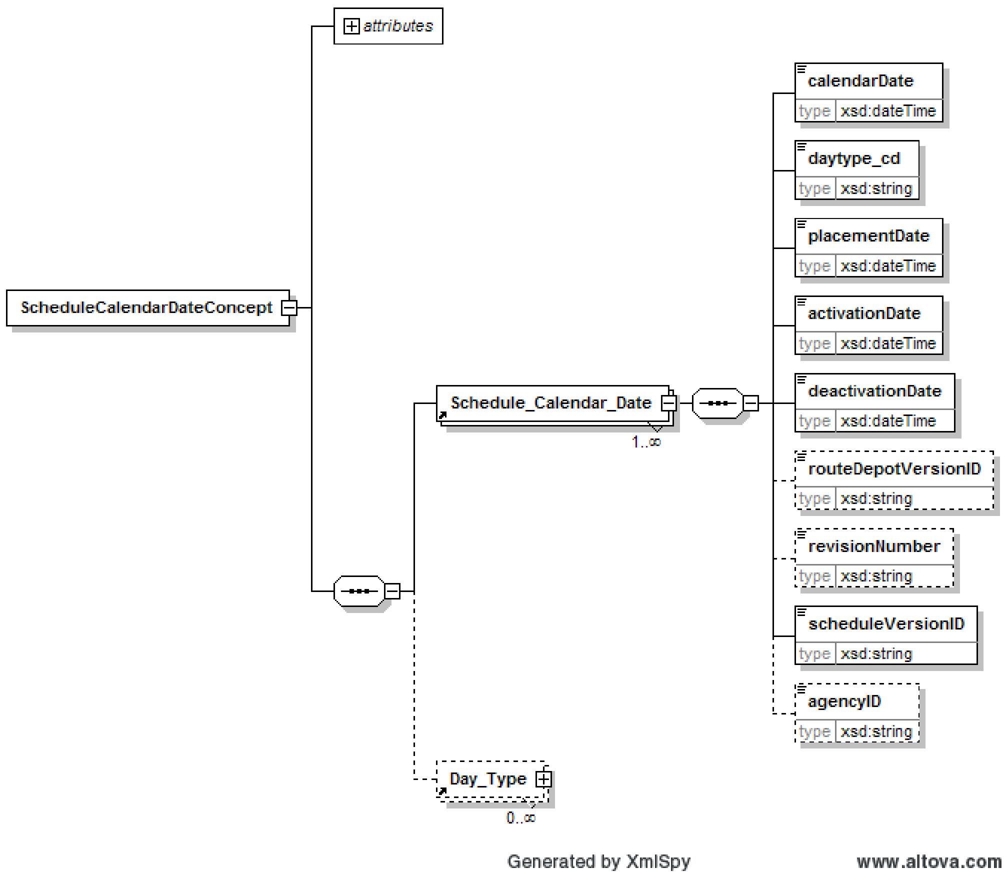


Figure 3-7: Schedule Calendar Date Element in the SCD XML Schema

Chapter 4: Agency Registration Branch Data Concepts

In This Chapter

- Understand the requirements related to the Agency Registration Branch and its major elements.
- Discover how Agency Registration branch and related data concepts are used.
- Learn how to apply the elements in the Agency Registration.

Agency Registration

The agency registration branch includes the following major elements:

- Agency
- Schedule Revision and Route Depot Version
- Route
- Organization Unit
- Depot
- Route Grouping
- Day Type

Purpose of a Agency Registration Branch Model

The Agency Registration Branch includes elements that define the version and history of the schedule data presented in the SDP XML Document by the submitting organization. This chapter:

- Describes high level information about the schedule including elements that define the version and routes included in the SDP XML Document;
- Defines the agency and organizational type elements, such as Organization Unit or Depot, which refer to the organizational units that are related to defining the version of the operational schedule contained in the SDP XML Document. For example, NYCT Bus generates and versions its schedule by route-depot. To this end, definition of the route and depot are included in the registration branch.

Agency Registration Conceptual Data Reference Model (CDRM) Description

The Agency Registration CDRM describes the relationships among the entities in the General Agency information (or the Agency Registration branch of the XML Schema). Figure 4-1 describes the overall relationships among the data concepts that are included in the General Agency Information CDRM. The model is described by the following indented paragraphs:

> "An Agency may generate one or more versioned schedules (Schedule_Version). An Agency may operate one or more Modes of service and an Agency may be composed of one or more Organizational_Units that compose Schedule_Versions. Moreover, the Agency may provide various ways in which customers may contact it (Additional_Contacts).
>
> "A Schedule_Version may contain more than one Schedule_Revision. The Schedule_Revision is a set of Routes (which may be grouped by Depot or garage, or by a specific Organization_Unit) which are associated with it. These Route_Depot_Versions are characterized by their activation and deactivation date. The Route_Depot_Version activation/deactivation dates should fall within the Schedule_Revision activation/deactivation dates which in turn should be contained within the Schedule_Version activation/deactivation dates.

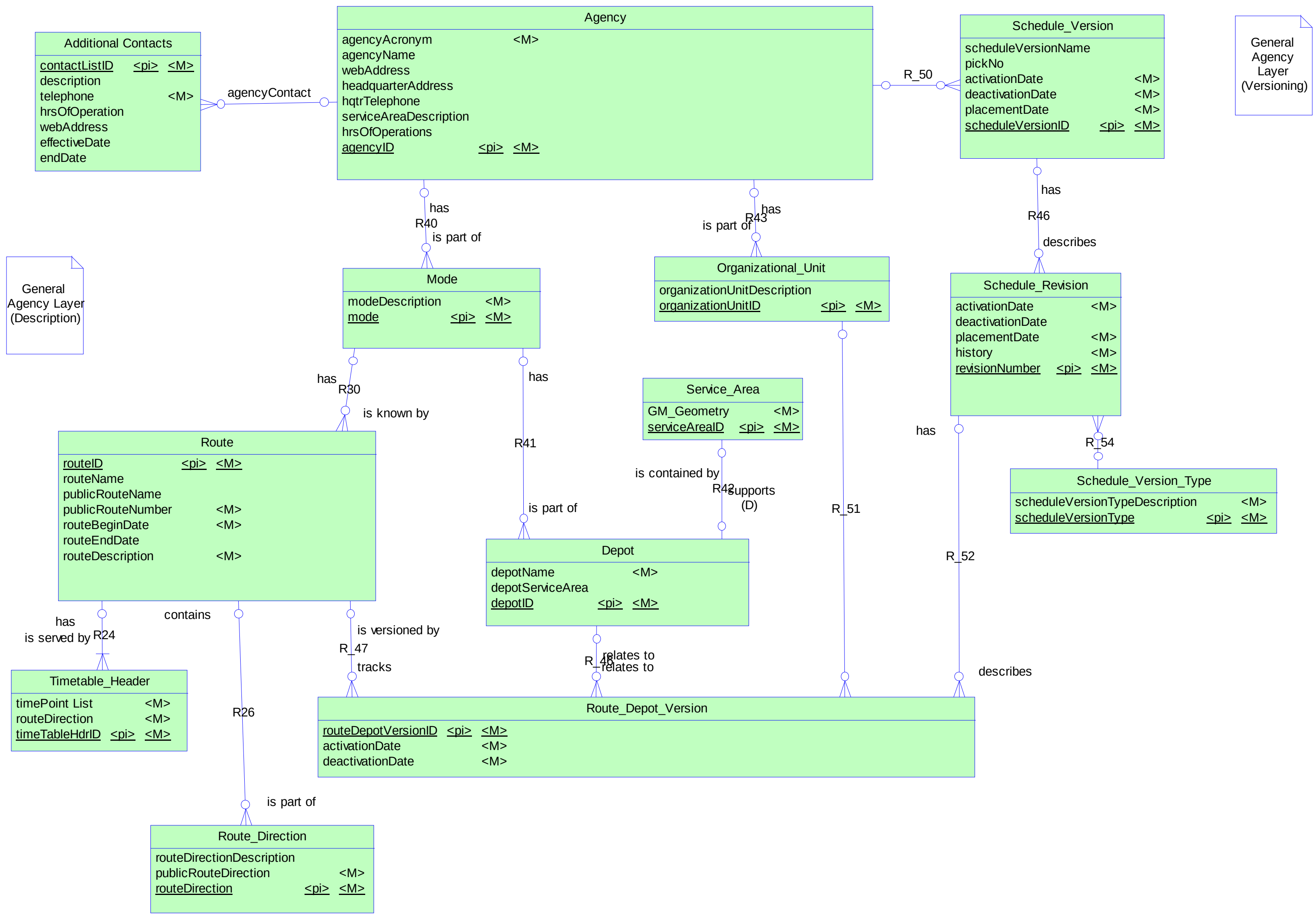


Figure 4-1: Agency Registration Conceptual Data Reference Model (Without Route Grouping)

"A Route has a public name and number known to the public, as well as a unique designator that is used for processing. The Route also includes a set of Route_Directions and may define a header form for use in a public timetable. The Timetable_Header is typically an edited (limited) set of events that occur across all patterns in the same direction. The header is used to describe the columns of the public timetable. The header may include the public location name or transit stop name, and a stop or location identifier."

"The Route_Grouping entity describes a collection of Patterns or Trips that are grouped together for a particular purpose (Route_Grouping_Type). A set of patterns or trips may be grouped together operationally in order to closely coordinate their scheduled headways along a common alignment or carriageway. Public timetables group routes together to communicate the frequency of service of routes that share a common corridor before they branch.). [Figure 4-2 shows the relationship of entities used to describe Route_Grouping.] A Route_Grouping may group trips of more than one Route (e.g., Long Island Bus Route N20 and N21), or only include selected patterns from a Route, for example only the express trips for origin to destination."

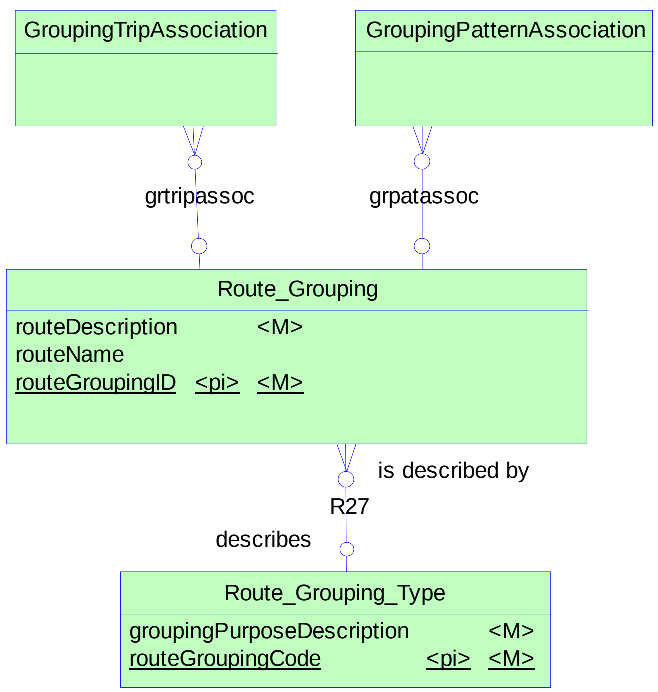


Figure 4-2: Route Grouping CDRM

SDP XML Schema Agency Registration Description

The CDRM's transformation of the General Agency entities to the XML Schema Agency Registration Branch levels most of the entities to a flatter organization. Figure 4-3 depicts the Agency Registration branch from the SDP XML Schema.

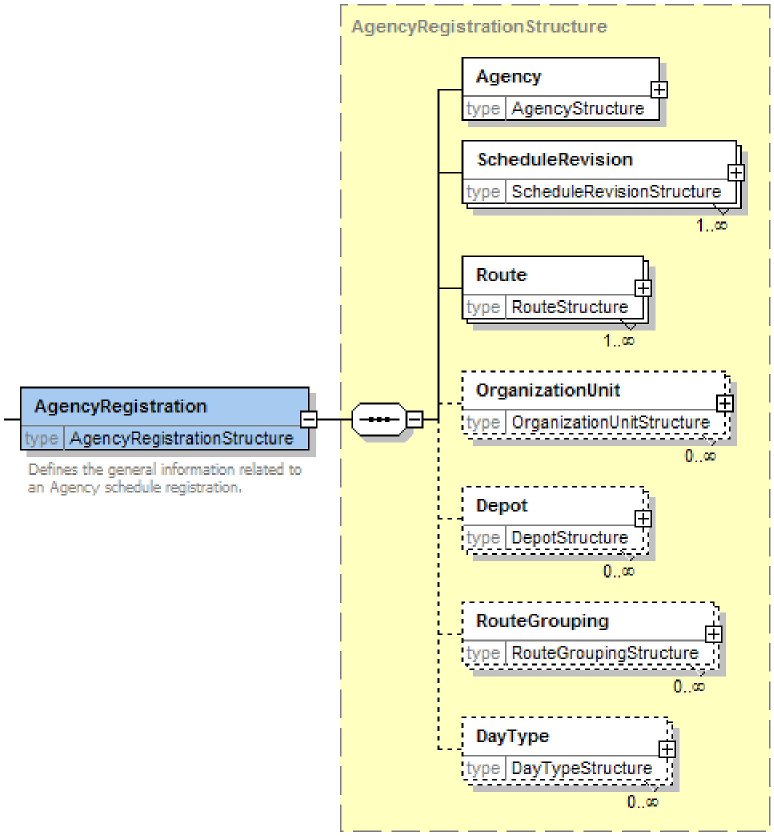


Figure 4-3: SDP XML Schema Fragment for the AgencyRegistration Branch

Although the CDRM includes Schedule_Version, it is not included in the Agency Registration Branch because the information is included as part of the SDP XML header attributes. Schedule Version attributes are discussed in more detail in Section 4.2.

Further Discussion and Relationship to SDP XML Schema

The data concepts depicted in the Agency Registration CDRM will be discussed in more detail later in this chapter, as well as key points pertaining to the CDRM's relationship to the SDP XML Schema.

Sections 4.1 through 4.4 describe the high level elements listed in the Agency Registration Branch, specifically (4.1) Agency, (4.2) Schedule Version, (4.3) Route and Route Direction, and (4.4) Organization Unit and Depot. Each section describes:

- Detailed functional requirements for the CDRM's data concepts.
- Additional rules applied (beyond those defined in Chapter 3) to migrate CDRM data concepts to the SDP XML Schema.
- Specific examples, such as showing the reader how business rules relate to translating native schedule data sets to the SDP XML Schema.

Section 4.1: Agency and Related Data Concepts

In This Section

- Learn the Agency Data Concept requirements and issues.
- Learn techniques on how to apply your native data to the Agency data concept.

Agency Definition
A transit agency is an organization that provides transportation services by bus, rail or other conveyance to the general public or special services on a regular, continuous basis.

Requirements for Agency Data Concept

Agency information needed by key downstate customer information applications typically is not stored in the schedule data. Unlike a single agency generated source or application, a regional source or application must differentiate among many agencies. To ensure that the information describes an agency according to its own policies, the SDP XML Document submittal requires current information pertaining to the organization that generates the schedule data set. Agency information may be inserted into a file once and read each time an SDP XML Document submittal is made. The basic agency information needed by various agencies and applications must describe the following:

- Agency name (doing business as)
- How the public refers to the agency
- Public contact for the agency
- Customer information contact channels (telephone, email, web site)
- Other channels/contacts for different types of customer information (e.g., service information, fare media purchase, lost and found, etc.)
- Hours of operation for service
- Geographical area covered by the agency's service
- Types of public transport (fixed) service (modes)

The SDP XML Schema makes provision for all these types of data, although they are not all required. Their status as either optional or mandatory is described in Table 4.1-2.

Specific requirements that drive the SDP XML Schema are listed in Table 4.1-1.

Table 4.1-1: Agency Requirements

#	Category	Requirements
1	Name and Identity	• Agency name and affiliation. • Description (may include a reference or incorporate documents on the operational statistics, inventory and performance measures, such as contained on MTA website "MTA Facts").
2	Contact	• Web address (e.g., http://beelinebus.westchestergov.com/). • Headquarter address/telephone number, for example: The Department of Transportation is located at: 100 East 1st Street Mount Vernon, N.Y. 10550 (914) 813-7777 between 7 a.m. to 7 p.m. weekdays TTY for the Hearing impaired is: (914) 813-7711 • Agencies may also support additional customer service contact numbers, e.g., customer information and lost & found.
3	Modes	• The modes offered by a transit agency, for example, bus, subway, light rail (LRT), commuter rail or ferry.
4	Service Area and Operations	• Narrative description of service area. • Hours of operation/schedule day. • These characteristics may differ based on mode, division, depot or organizational unit. For example, Coach has several motorbus services assigned to different areas; NYCT operates different modes, routes and schedule versions from different depots.
5	Operations Association With Other "Roles"	• Operations (and thus schedules) may be associated with specific organizational units, for example, contractor may issue schedules for the routes they service. An agency should be able to include a reference to an internally designated organizational unit. • Organizational unit (user defined identification and description).
6	Schedule Owner and Schedule Version Custodian	• Within the context of the SDP, the transit agency may designate the "owner" of a schedule. • In some cases, there are schedule version custodians that are defined in order to distinguish schedule versions. These custodians may also be associated with an agency.

Conceptual Data Reference Model Description for Agency

The agency-related requirements may be described in the CDRM depicted in Figure 4-4.

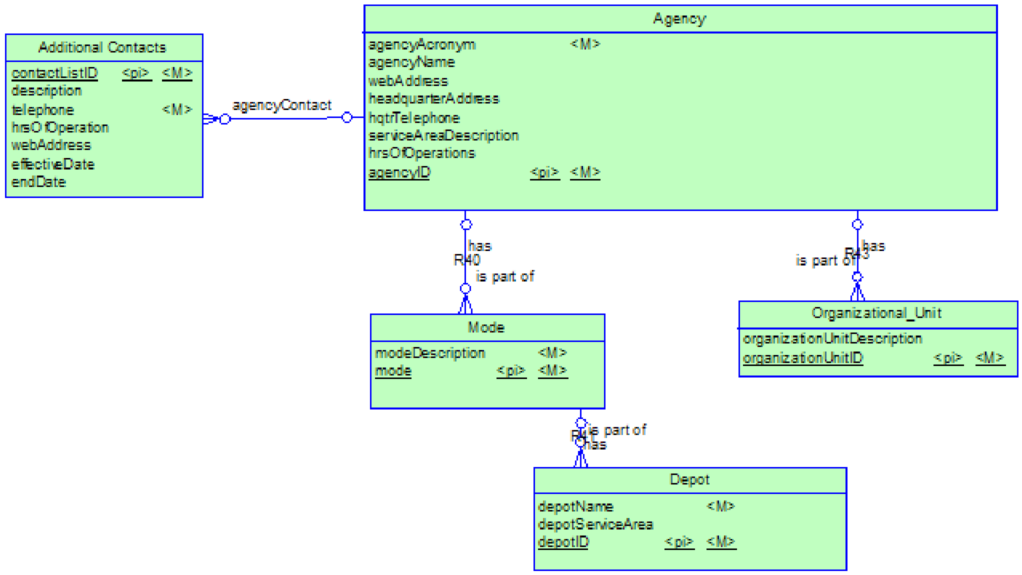


Figure 4-4: Agency Entity in the Conceptual Data Reference Model

The CDRM is described by the following:

> An Agency offers one or more Modes of service. Each Mode may require one or more Depots (garages, divisions or yards) to meet the operational demand in the area serviced by the Depot. In addition, an Agency may support one or more Organizational_Units[6] that are responsible for providing service. An Agency may support multiple customer service contact channels (Additional_Contacts), such as customer information and lost & found.

Note: A shortcut is used to define the agency address for the Agency entity in Figure 4-4. It is defined as an Address element in the XML Schema and an entity in the physical database implementations.

High Level Agency XML Schema Model Description

The Agency element in the SDP XML Schema Model includes nine child elements, two of which reference other complex type elements (i.e., headquarterAddress and agencyContact). The Agency fragment from the SDP XML Schema is depicted in Figure 4-5. In implementing the CDRM into the XML Schema, a number of rules were used.

[6] This is a restrictive definition of “organizational unit”. See Business Rule in next Section which discusses the scope of this data concept within the reference model.

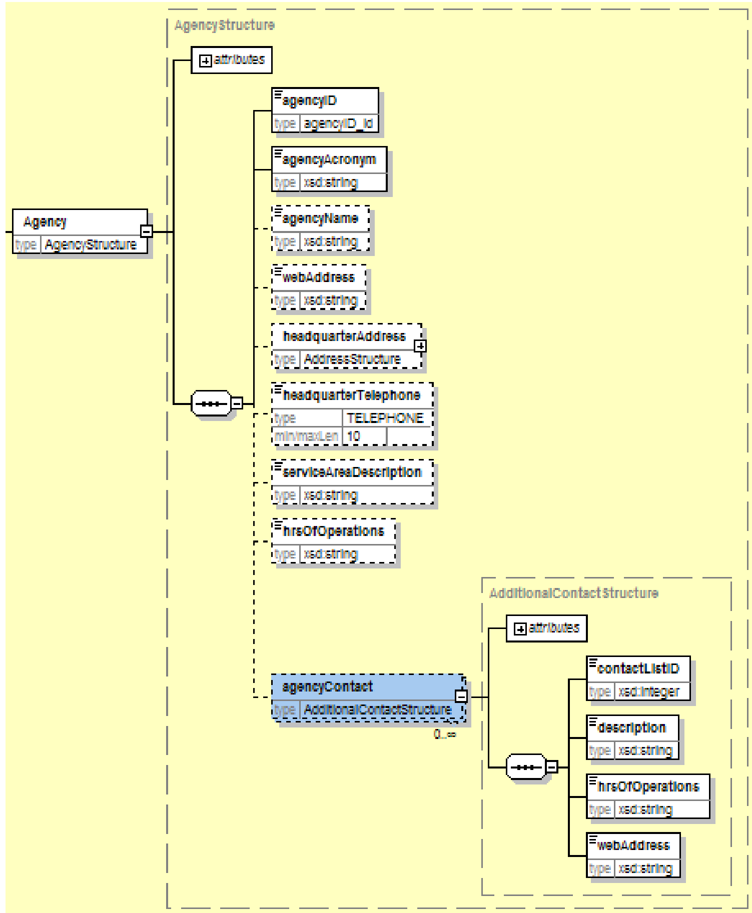


Figure 4-5: Agency Element in SDP XML Schema

- As a multi-element data type, the headquarterAddress element references the AddressStructure, an embedded complex type.
- The Additional_Contacts is also a child element of Agency. It is tagged in Agency as agencyContact.
- The "attribute" notations attached to Agency and agencyContact consists of an effectiveDate and endDate. The effectiveDate refers to the placement date of the record. The endDate refers to the date the contact or agency information is obsolete or has been superseded. This artifact supports archiving and repository update functions. These attributes were assigned to most XML elements even if the requirement was not explicitly stated in the CDRM.
- The agencyID is unique among all agencies, and thus should be assigned by the regional repository.
- The contactListID is an arbitrary index to distinguish each agencyContact.

Detailed Agency Data Descriptions and Guidance

This section provides data type information and guidance associated with the Agency and Additional Contacts elements described above. The guidance for each element is consolidated into a table with the following column headings: Requirement status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0.

A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML document deployment.

Table 4.1-2 incorporates Agency Guidance and Table 4.1-3 incorporates Additional Contact Guidance.

Table 4.1-2: Agency Guidance

	Element Name	Type	Questions to Ask
M	agencyID	agencyID_id UNIQUE	This designator is assigned by SDP/TSDEA registration. (if not available, use agencyAcronym.)
M	agencyAcronym	String	Acronym that is known by the public, for example LIRR or MTA LIRR.
O	agencyName	String	Full name of agency, for example “Metropolitan Transportation Authority New York City Transit” or “MTA New York City Transit.”
O	webAddress	String	The universal resource link or home page web address of the agency.
O	hdqtTelephone	TELEPHONE	The general information number or a general customer information telephone number. Format should be: aaa-ttt-tttt (where aaa refers to the area code followed by seven digits).
O	headquarterAddress	ADDRESS	See address structure guidance in Part 3 SDP Schema Guidance Template.
O	serviceAreaDescription	String	A description of service area. The description is used to present to the public. As such the description should avoid abbreviations and excessive capitalization.
O	hrsOfOperations	String	A basic description of the hours of operation, for example: “5:30 a.m. to 10 p.m.” or “24 hours.”
O	agencyContact	Additional Contacts	Are there other places where customers may contact the agency? (e.g., customer help line, lost and found, complaints, etc.). If yes, then complete a new Additional Contact element for each the contact type.
O	@ effectiveDate	date	[attribute] format: yyyy-mm-dd
O	@ endDate	Date	[attribute] format: yyyy-mm-dd; if no end date, use 9999-12-31

Table 4.1-3: Additional_Contact Guidance

	Element Name	Type	Questions to Ask
M	contactListID	Autonumber	Assign a unique number for each Additional Contact.
M	description	String	Describe the type of contact (e.g., lost and found).
O	telephone	TELEPHONE	The contact information number or a general customer information telephone number. Format should be: aaa-ttt-tttt (where aaa refers to the area code followed by seven digits).
O	hrsOfOperations	string	The hours of operation for the contact information, for example: if a customer service representative telephone line is open from "5:30 a.m. to 10 p.m." or "24 hours."
O	webAddress	String	The universal resource link or home page web address of the agency.

Example for Populating the Agency Structure

Table 4.1-4 provides an example of how Agency elements might be submitted by the Long Island Rail Road. For training purposes, the table includes the element's status as mandatory or optional, the element name, the element definition and the data value from LIRR. The headquarter's address is not listed, which is acceptable, since it is an optional element that is not needed at the moment by any applications. The "pi" implies that the element is a **p**rimary **i**dentifier, and is unique and mandatory.

Table 4.1-4: Agency Element Definition and Examples

	Element	Definition	Example
pi	agencyID	A unique identifier that designates a transit operator.	300
M	agencyAcronym	An abbreviation for the Transit Agency's name that is known by the public	LIRR
O	agencyName	The complete name of the Transit Agency	Long Island Rail Road
O	webAddress	The universal resource location (URL) for the Agency's web site	www.lirr.org
O	headquarterAddress	The physical address for the Transit Agency's headquarter facility	
O	headquarterTelephone	The telephone numbers that may be used to contact the Transit Agency headquarter facility.	718-217-5477 (LIRR) 516-822-5477 (LIRR) 631-231-5477 (LIRR)

Table 4.1-4: Agency Element Definition and Examples

	Element	Definition	Example
O	serviceAreaDescription	A short description of the service area covered by an Operator.	"The LIRR system is comprised of over 700 miles of track on 11 different branches, stretching from Montauk–on the eastern tip of Long Island–to the refurbished Penn Station in the heart of Manhattan, approximately 120 miles away. Along the way, the LIRR serves 124 stations in Nassau, Suffolk, Queens, Brooklyn and Manhattan."
O	hrsOfOperation	The hours of operations of revenue service as presented to the public. The maximum number of hours is 24.	24

Agency Example From a SDP XML Document

A typical example, which incorporates agency information, from a SDP XML Document with dummy data is listed below:


```
<Agency endDate="1967-08-13" effectiveDate="1967-08-13">
        <agencyID>String</agencyID>
        <agencyAcronym>String</agencyAcronym>
        <agencyName>String</agencyName>
        <webAddress>String</webAddress>
        <headquarterAddress>
                <addressID>String</addressID>
                <addressSegID>String</addressSegID>
                <recordDate>1967-08-13</recordDate>
                <addressNumber>0</addressNumber>
                <directionPrefix>0</directionPrefix>
                <typePrefix>String</typePrefix>
                <streetName>String</streetName>
                <typeSuffix>String</typeSuffix>
                <directionSuffix>0</directionSuffix>
                <completeName>String</completeName>
                <unitType>0</unitType>
                <unitDesignation>String</unitDesignation>
                <secondLine>String</secondLine>
                <postalCommunity>String</postalCommunity>
                <postalState>0</postalState>
                <postalCode>String</postalCode>
                <status>0</status>
        </headquarterAddress>
        <headquarterTelephone>aaaaaaaaaa</headquarterTelephone>
        <serviceAreaDescription>String</serviceAreaDescription>
        <hrsOfOperations>String</hrsOfOperations>
```

```xml
        <agencyContact endDate="1967-08-13" effectiveDate="1967-08-13">
                <contactListID>0</contactListID>
                <description>String</description>
                <hrsOfOperations>String</hrsOfOperations>
                <webAddress>String</webAddress>
        </agencyContact>
        <agencyContact endDate="1967-08-13" effectiveDate="1967-08-13">
                <contactListID>0</contactListID>
                <description>String</description>
                <hrsOfOperations>String</hrsOfOperations>
                <webAddress>String</webAddress>
        </agencyContact>
</Agency>
```


An example of Agency information that might appear in the NYCT Bus SDP XML Document is:


```xml
<Agency effectiveDate="2006-11-01" endDate="9999-12-31">
  <agencyID>100.2</agencyID>
  <agencyAcronym>NYCT</agencyAcronym>
  <agencyName>New York City Transit Bus</agencyName>
  <webAddress>http://www.mta.nyc.ny.us/nyct/bus/index.html</webAddress>
</Agency>
```

Section 4.2: Schedule Version and Related Data Concepts

Schedule Version Definition
The time period, described by the start date and time and optionally end date and time, when an agency service provision is valid. Due to the nature of transit schedules, different routes, depots and organizational units may implement various versions that operate during the same time period.

In This Section

- Learn about the Schedule Version CDRM and its relationship to the SDP XML Schema.
- Discover how schedule version and related data concepts are used.
- Learn how to apply Schedule Version, Schedule Revision and Route Depot Version.

Typical Schedule Version Practice

The scheduling process follows a typical cycle no matter the locale, mode or organization. As a human resource-driven service, a transit schedule release (when all the schedule components are checked and approved) is typically driven by the operators' selection of work or "pick." The operator pick typically has an effective start and end date of service that spans three, four or sometimes six months. Different organizations have different naming conventions for pick numbers, for example, many concatenate the last two digits of the year with a number referring to a pick period that occurs during the year, *305* for the Fall (3rd) pick in 2005, others may use the season as an indicator of the pick, e.g., Fall 2005. Pick numbers define the high level *schedule version identifier*. As such, the pick period number does not change even if elements of the schedule do change.

Schedule Revision
Records the state and manages the changes to the designated schedule version.

Route Depot Version
Characteristics of the schedule version that applies to a route that operates from a specific depot, or that is generated by a specific organizational unit. In the case where there is no distinction on specific organizational unit, the depotID or organizationUnitID is a non-identifying foreign key.

A *revision number* may indicate that one or more changes were made to various components of the schedule version. Examples of revisions that affect scheduled service are listed below:

- A construction project starts during the middle of the schedule period and service must be changed to accommodate a detour or delay.
- A trip starting time (and possibly all subsequent trip times) was changed to start two minutes later due to increased running times.
- Special holiday service is initiated two weeks before the holiday every weekday.

Many agencies implement schedule revisions without altering the original pick number because an error was found in the data set. In such cases, the data set may be completely replaced with a new revision. The original data set will not be kept or archived due to the error. Yet it is still judicious to track the revision or generation version that is used. To this end, the Schedule Revision keeps track of the changes made to all or part of the schedule data set.

Some changes are only temporary and depend on environmental conditions, time of the year, or external circumstances. For example,

- A change may occur only during a specified period of time and only affect a few trips or trains, e.g., the five weeks before Christmas;

- An exception may apply to an entire route or patterns of a route, e.g., major construction that was not incorporated into the Pick schedule;
- An exception may describe the extra trains that are needed on a periodic basis, for example, when the Mets play at Shea stadium at 5 p.m. versus when they play at 7 p.m.; or
- A template for emergency service operations for severe weather conditions (a.m./p.m./mid-day evacuation) is another type of exception based schedule.

Temporary changes are not well documented in current industry practice. They may be documented as an exception in a special file, as is typically the current practice. The temporary change may be associated with an entire set of route schedules, or designated within a standard schedule as specific “special” day types, service keys, or as extra or exception service that run on certain days.

Different modes reduce their schedule exception types to different levels of resolutions. For example, rail systems, because they run fewer trains than bus trips may schedule their service and exception service on a train by train basis, while bus service may package their schedule and exception service on a route level. Rail systems plan physical changes to the location of their stops several years in advance, while bus stops on average change 10 percent annually.

Requirements for Schedule Version Data Concept

The requirements that drive the Schedule Version Data Concept are described in Table 4.2-1.

Table 4.2-1: Requirements for Schedule Version and Related Data

#	Category	Requirements
1	Schedule version coverage and identification	• A schedule version number is used to group all the schedule components for a specified period of time (activation to deactivation dates). The number should not change over that period (although the deactivation date may be changed). • The schedule components related to a specific schedule version may be submitted to the TSDEA in separate SDP documents. The TSDEA can accommodate changes to the schedule only if all the submittals include a common schedule version number. • A schedule version applies to all the components of an agency's schedule that are active during a schedule version period. Those components include routes/branches/lines, trips/trains, patterns and those elements that compose trips and patterns (e.g., timepoints, stop points, notes). • A unique identifier or description (e.g., time period) describes a schedule version for each Transit Agency at the level a schedule component is versioned. • The schedule version is typically related to the schedule components from which operator's pick their work assignments over the proposed operational period. For that reason, the schedule version may be the same as the "pick" number. • A schedule version description shall also include the status of the schedule version including all revisions.

2	Schedule Types	A schedule may be categorized as follows:
		• ***Original***: the base schedule assigned to the schedule version number.
		• ***Revision:*** a change to part of a schedule for a specified period.
		  - *Permanent (revision):* a change to part of a schedule whose duration may be after the activation date, but whose deactivation date is the same as the original schedule version. Note: a temporary revision that overlaps two schedule version periods may be considered a permanent change to the first schedule version.
		  - *Temporary (revision):* a change to part of a schedule whose duration may fall within the original schedule version period, or for non-sequential days that fall within the original schedule version period.
		• ***Suspended:*** A schedule version that is no longer valid. A suspended schedule version renders all schedule components and files that use the schedule version number as invalid.

3	Packaging Schedule Components	At a minimum, schedule components shall be packaged at a route/depot level in an SDP document. This may imply that every route is contained in a separate package. This requirement does not restrict an agency from submitting an SDP document with its entire schedule for all routes.

Table 4.2-1: Requirements for Schedule Version and Related Data

#	Category	Requirements
4	Rules for file submissions for original schedule version schedule components	An original schedule version submission must contain all the schedule components and related libraries that apply to that schedule version. The information is the baseline information. Once a schedule version is suspended, all data pertaining to that schedule version is removed from the TSDEA. • A set of integrity checks will be developed that validate the logical consistency of the data set as a whole.
5	Rules for including schedule components to a revision file	A revised schedule version submission must contain the affected ***route*** and all the route schedule components. If the change was part of the SDP library, then the selected library entries should also be included. • The permanent changes will be applied to the previous, valid schedule. This will entail replacing the entire route description and related entities. Integrity checks will be used to validate the logical consistency of the revised data set as a whole (all routes and schedule components). When the data set passes the tests, the permanent changes will be incorporated into the production level data set. If not, the resulting errors will be detailed and sent to the data producer. • Temporary changes will be stored in a separate file and apply to specific dates in the service calendar. The SDP documents that contain temporary schedules should be self contained and include all applicable SDP library entries.
6	Rules for schedule version suspensions	A SDP document that is designated as a suspended schedule version does not need to contain any schedule components or libraries. As a result of its receipt, all data and files related to the schedule version shall be purged from the TSDEA.
7	Deactivation date	• A schedule version need not specify the deactivation date when the scheduled service is first activated. The most significant date available (on most computing platforms is 31 December 9999) should be used to designate the schedule version deactivation date if it is unknown. The SDP should be updated with the closing date when it is known. • When a deactivation date is updated by the SDP registrant, the TSDEA will be responsible for updating all SDP document's schedule components.

Schedule Version Conceptual Data Reference Model Description

A Schedule Version defines the activation time period of a schedule. Specifically, a scheduleVersionID is a unique identifier associating the collection of service provision components that are valid for a period of time. The schedule version is typically defined as the services selected by operators as their work assignments during a period during the year. Different organizational units or operator/depot units may generate the schedule components to form a mode-, organization- or route/depot-based schedule version. A revision number will track changes to the service provision specifically the routes associated with an organizational unit or depot. Figure 4-6 shows the CDRM for Schedule Version.

A Schedule Version is generated once a scheduling period. (The Schedule Revision associated with an original scheduleVersionType defaults to zero.) An original schedule and scheduling components (i.e., routes as generated by either a depot or organizational unit) may be revised several times over its activation period, and is assigned a unique, sequential revision number for each revision. Depending on the type of revision, the Schedule Revision version may have a different activation and deactivation date for multiple route schedules. A Schedule Revision version that is of type “temporary” should include the valid dates of operation and be associated with the Service Calendar Date. [The schedule calendar date and day type data concepts are described in greater detail in Chapter 9: Schedule Calendar Date Data Concept.]

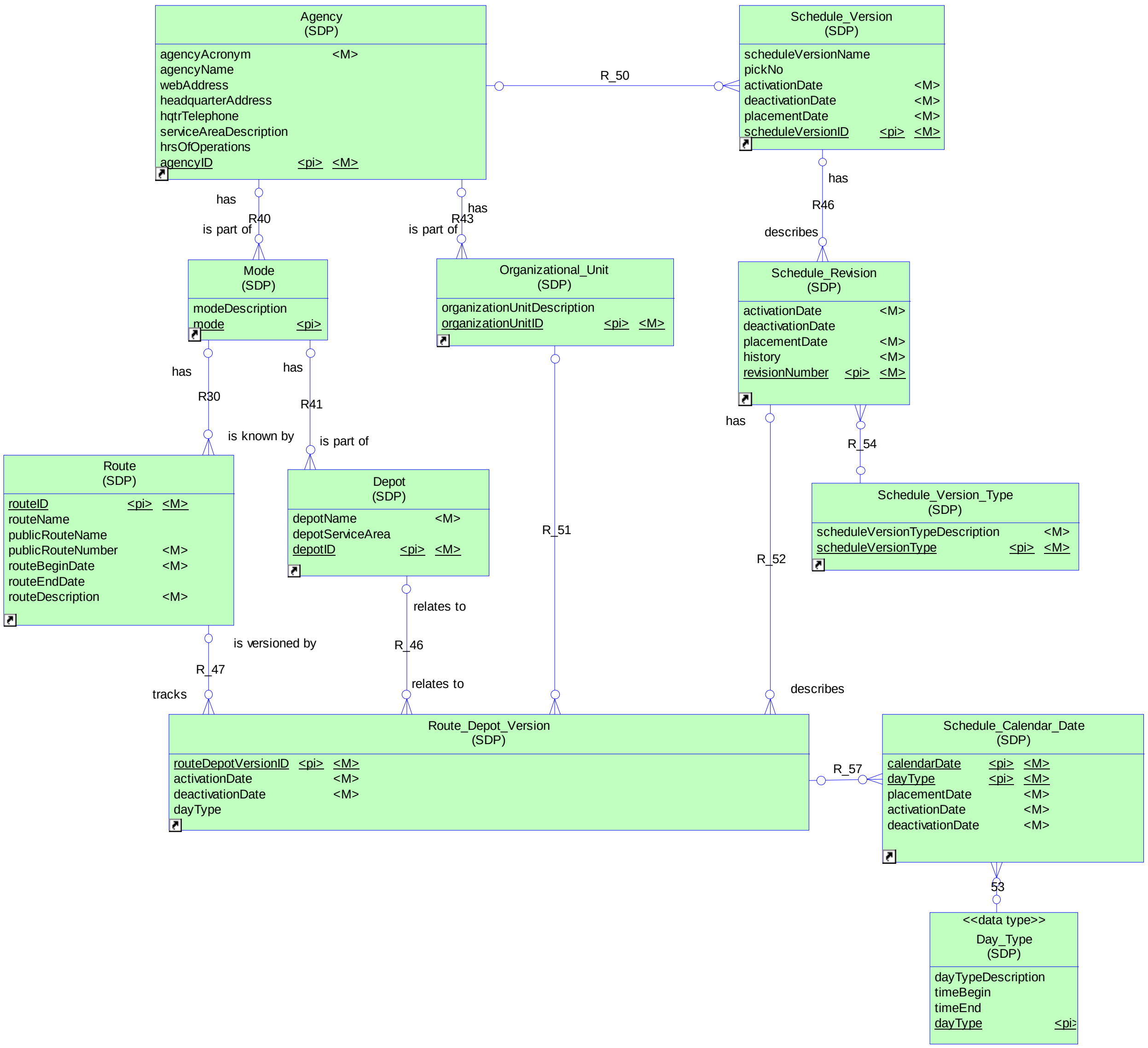


Figure 4-6: Schedule Version Conceptual Data Reference Model

XML Schema Descriptions for Schedule Version and Revision, Route Depot Version Elements

There are specific assumptions and requirements associated with the Schedule Version and Schedule Revision elements that are migrated from the CDRM to the SDP XML Schema. These requirements are described in the following subsections.

<u>Schedule Version</u>. The Schedule Version concept is implemented in the SDP XML Schema as an attribute group.[7] As such, an SDP XML Document represents only one schedule version of a single agency. Schedule Version attribute group contains six attributes. Although optional attributes are present, they ***must be*** included in the SDP Document header section (even if left empty) for validation. An illustration of the attribute group may be viewed in the top right hand box in Figure 4-7.

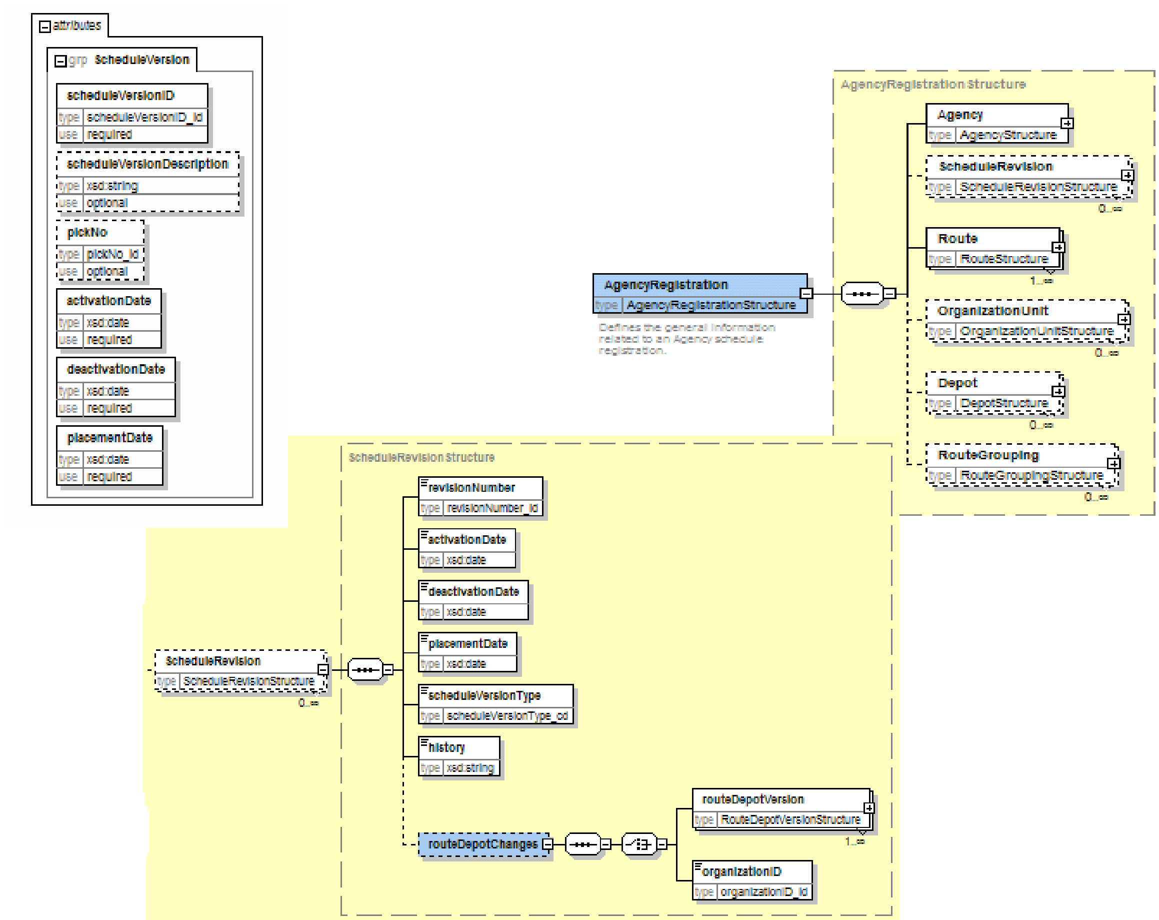


Figure 4-7: Schedule Version and Schedule Revision Elements from SDP XML Schema

<u>Schedule Revision</u>. The Schedule Revision is represented as an element in the Agency Registration branch. Zero or more Schedule Revisions may be included in the document. The element contains seven elements, the first six are mandatory and the last element, routeDepotChanges, is optional. The routeDepotChanges element embeds the Route Depot Version elements or the associated organization unit (organizationUnitID) that generates or changes the schedule.

The routeDepotChanges is an optional element, which, when included, must be composed of one or more Route Depot Versions or a single organizationUnitID. The routeDepotChanges is inserted when:

---

[7] Attribute and Attribute Group as defined by the XML Schema standard.

- An Agency assigns version numbers to each route or group of routes generated for a specific depot (routeDepotVersion elements are required) or organizational unit (organizationID is required), or
- An agency submits a partial update of their schedule. Then routeDepotVersion elements are submitted for each route that was changed under this revision number and is included in the file.

The first SDP Document submission for service should include Schedule Revision where schedule version is an "original" scheduleVersionType and the revisionNumber is equal to "0."

Figure 4-8 shows excerpts of elements of Schedule Revision and RouteDepot Version data concepts in the SDP XML Schema. These elements are illustrated in the bottom yellow shaded box.

Route Depot Version. As mentioned in Schedule Revision, the Route Depot Version element is referenced as one or more elements in the routeDepotChanges element of the Schedule Revision element. The element, as illustrated in Figure 4-8, should be included when the SDP consists of only a partial list of routes, an update of a select number of routes, or when an Agency groups its routes by depot. For example, the NYCT will register its data by route depot version in order to group its routes by depot and day type.

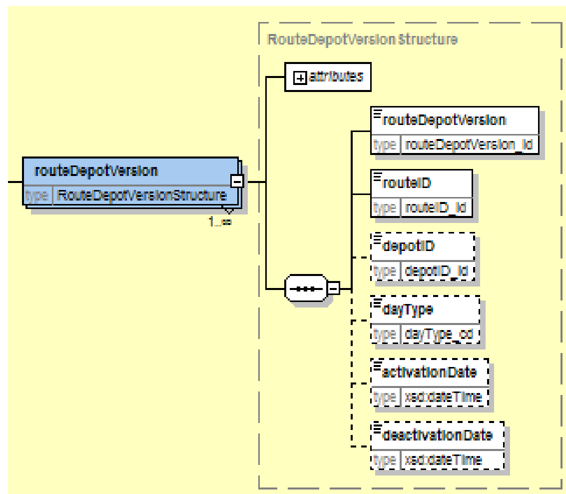


Figure 4-8: Route Depot Version Element in the SDP XML Schema

The Route Depot Version consists of only two mandatory elements, routeDepotVersion and routeID. Inclusion of the Route Depot Version structure provides a listing of the routes that are included in the document.

The activation and deactivation dates, although not required, may be needed to ensure that the routeDepotVersion is consistent with the Schedule Version and Revision dates. For example, the route depot version dates should be bounded by the revision dates and the revision dates should be bounded by the schedule version dates.

Furthermore, there are cases when changes to a route result in a Route element with the same designator appearing more than once in a data set. For example, the patterns (and consequently trips) associated with a route may change for a week due to construction on the route pattern. As such, the routeDepotVersion activationDate and deactivationDate should be included and match the Route element routeBeginDate and routeEndDate to associate the correct Route and Route Depot Version.

Schedule Version Detailed Data Descriptions and Guidance

This section describes the format and guidance associated with the ScheduleVersion attribute group and ScheduleRevision and RouteDepotVersion elements in the data concepts described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML document deployment.

The following tables provide data descriptions and guidance:

- Table 4.2-2 addresses Schedule Version Attribute Group Guidance
- Table 4.2-3 addresses Schedule Revision Guidance
- Table 4.2-4 addresses Route Depot Version Guidance

Table 4.2-2: Schedule Version Attribute Group Guidance

	Attribute Name	Type	Questions to Ask
M	scheduleVersionID	scheduleVersionID_id	Is there a special identifier used to designate a schedule?
O	scheduleVersionDescription	string	Is there a special field that describes the schedule version?
O	pickNo	string	What operator sign up or pick number was this schedule associated with? This element may be the same as the scheduleVersionID.
M	activationDate	date	When does this schedule begin? [yyyy-mm-dd]
M	deactivationDate	date	When does this schedule end? If unknown then use default value of 9999-12-31.
M	placementDate	date	Date on which schedule was generated or developed (if no date is available then use date compiled into SDP format).

Table 4.2-3: Schedule Revision Guidance

	Element Name	Type	Questions to Ask
M	revisionNumber	revisionNumber_id UNIQUE	If there is no revision number this may be set to "0" and incremented whenever a change is made.
M	activationDate	date	If corresponds to schedule version, then it is assigned the same date; if this version is temporary (for example a long term detour routing), then start date of revised schedule is used.
M	deactivationDate	date	See activationDate comment. If unknown then use default value of 9999-12-31.
M	placementDate	date	Date schedule revision was generated or placed. Use date of SDP document generation if the revision date is not known.
M	scheduleVersionType	scheduleVersionType_cd	Codes include [original, rev-permanent, rev-temporary, suspended].
M	history	string	Use a standard convention for generating the history, for example “1-original; 2-updated routes 2, 10, 12;” etc.
O	routeDepotChanges	CHOICE of organizationID or Route Depot Version	If the element is included, one of these elements must be included. The organizationID may refer to an arbitrary, unique identifier that designates an organization. Because this is a choice element, the organizationID and one or more routeDepotVersion elements are wrapped in <routeDepotChanges> tags.

Table 4.2-4: Route Depot Version Guidance

	Element Name	Type	Questions to Ask
M	routeDepotVersion	routeDepotVersion_id UNIQUE	If there is no version number to trace to at the agency (e.g., file name for STIF/RTIF files, or route version), then assign unique sequential number.
M	routeID	routeID_id	Unique identifier for routeID that is used throughout the document. This identifier is validated against the set of primary routeIDs in the Route element.
O	depotID	depotID_id	Although a unique identifier linked to Depot.depotID, this identifier is not needed unless the route number and set of trips are unique only for the route associated with a specific depot or garage (e.g., like NYCT).
O	dayType	dayType_cd	Because many scheduling systems output data by route and day type, schedule data may be packaged in an SDP document by route and day type. The dayType element provides a means to trace the information in this document to a specific file.
O	activationDate	date	This is the date the route-depot version is activated. This field may be redundant with the Revision, Route or ScheduleVersion activationDate. If a single routeID is designated more than once in the data set because its service runs on mutually exclusive dates, then the activationDate must match the ROUTE element's routeBeginDate.
O	deactivationDate	date	Same as activationDate.
O	effectiveDate	date	[attribute name] use to record placement of record.
O	endDate	date	[attribute name] use to record when this record will become obsolete.

Examples of Schedule Version and Revision

This example illustrates how a progression of changes may occur over the schedule version lifecycle. The following bullets provide background for the example:

- Long Island Bus may submit 49 files that contain the original schedule components for the Fall 2006 version (306).
- Several routes may need to be changed in October due to coordination of specific routes with Long Island Railroad. These are contained in Revision 1.
- Later that month, damage occurs to a major arterial and the road is closed. Several routes will be affected for at least three weeks while the county deals with the problem. LIB revises the patterns, by removing certain stops and adding new ones at different locations.

- Several times later that year, transit stops are removed or added, and slight changes are made to the schedule.
- The routes and library entries that are affected are included in a SDP document and a new revision number is created to track the submittals. Finally, on 2 January 2007, the new schedule becomes active (107) and the existing schedule version 306 is removed.

The progression of changes to the schedule version (sign up) 306 is illustrated in Figure 4-9.

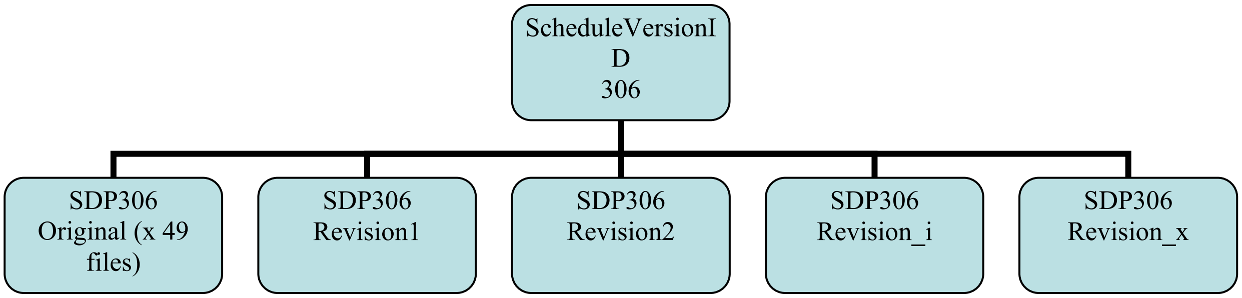


Figure 4-9: Example of SDP Schedule Version Submittals

For the suspension case, the schedule version number is withdrawn and may be reused by an agency. Assume that an agency produces and submits a schedule that must later be rewritten due to unforeseen circumstances. The entire schedule and schedule components may be purged from the TSDEA with the schedule version type "Suspend" code as shown in Figure 4-10. In order for the TSDEA to track this set of file submittals, temporal information such as activation and deactivation dates are needed, and should be associated with the description of the schedule version.

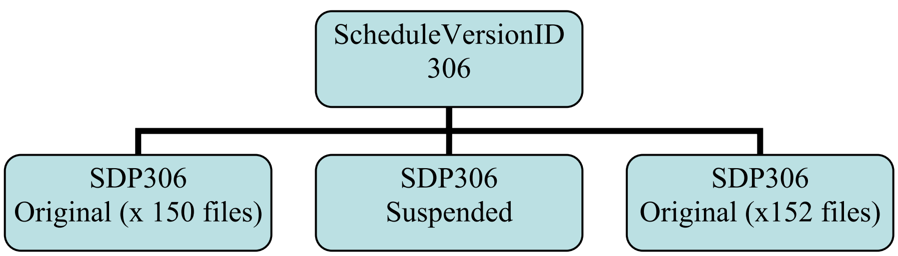


Figure 4-10: Example of SDP Schedule Version Suspension

Example of Typical Schedule Version Attribute Group

The Schedule Version identifier is stored in the header. An excerpt from the Long Island Bus SDP XML document might appear as follows:

```
scheduleVersionID="107"
scheduleVersionDescription=""
pickNo="107"
activationDate="2007-01-07"
deactivationDate="9999-12-31"
placementDate="2006-12-28"
```

Example of Typical Schedule Revision with routeDepotChanges

The NYCT Bus SDP XML Document will include a routeDepotChanges element for each file it includes in the SDP XML document. In this example, the Schedule Revision element is included with six routeDepotChange elements, one for each route-depot-day type for routes BX1 and BX2.


```xml
<ScheduleRevision>
 <revisionNumber>0</revisionNumber>
 <activationDate>2006-11-02</activationDate>
 <deactivationDate>2006-11-30</deactivationDate>
 <placementDate>2006-11-01</placementDate>
 <scheduleVersionType>original</scheduleVersionType>
 <history>1-original</history>
 <routeDepotChanges>
  <routeDepotVersion>
   <routeDepotVersion> BX1_206273</routeDepotVersion>
   <routeID>BX1</routeID>
   <depotID>KB</depotID>
   <dayType>weekday</dayType>
  </routeDepotVersion>
  <routeDepotVersion>
   <routeDepotVersion> BX1_206274</routeDepotVersion>
   <routeID>BX1</routeID>
   <depotID>KB</depotID>
   <dayType>sat</dayType>
  </routeDepotVersion>
  <routeDepotVersion>
   <routeDepotVersion> BX1_206275</routeDepotVersion>
   <routeID>BX1</routeID>
   <depotID>KB</depotID>
   <dayType>sun</dayType>
  </routeDepotVersion>
  <routeDepotVersion>
   <routeDepotVersion> BX2_206273</routeDepotVersion>
   <routeID>BX2</routeID>
   <depotID>KB</depotID>
   <dayType>weekday</dayType>
  </routeDepotVersion>
  <routeDepotVersion>
   <routeDepotVersion> BX2_206274</routeDepotVersion>
   <routeID>BX2</routeID>
   <depotID>KB</depotID>
   <dayType>sat</dayType>
  </routeDepotVersion>
  <routeDepotVersion>
```

```
      &lt;routeDepotVersion&gt; BX2_206275&lt;/routeDepotVersion&gt;
      &lt;routeID&gt;BX2&lt;/routeID&gt;
      &lt;depotID&gt;KB&lt;/depotID&gt;
      &lt;dayType&gt;sun&lt;/dayType&gt;
    &lt;/routeDepotVersion&gt;
  &lt;/routeDepotChanges&gt;
&lt;/ScheduleRevision&gt;
```

Section 4.3: Route and Route Direction Data Concepts

In This Section

- Learn about the Route Data Concept.
- Learn how to apply the elements in the Route data concept.
- Learn about the need for and application of the Timetable Header concept.

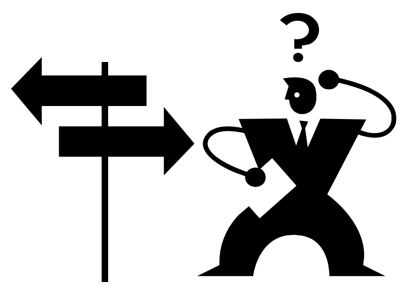
Route Definition
A collection of patterns and trips in revenue service with a common identifier or name.

Route Direction Definition
A description or name of the route direction that is recognized by the public.

This section covers the definitions for Route and Route Direction. The elements Route Grouping and Timetable Header are directly related to Route, but are advanced concepts that are covered in Chapter 10: Advanced Data Concepts.

Typical Route Definition Practice in Transit

The Route, described by the physical path and the traversal of service over that path, constitutes one of the key pieces of schedule information. The purpose of the SDP Route data concept (designated by a route identifier and name) is to describe a critical aspect of transit service primarily from the perspective of the customer or transit rider. As such, the content of the Route services is the collection of patterns and trips that service the customer, as opposed to the operator, scheduler or other internal staff. The Route Grouping data concept (Chapter 10: Advanced Data Concepts) includes details for developing a collection of patterns and trips that serve an alternative target audience, and may be known by a different common name or number.



Route direction is signified by a variety of sometimes overlapping methods across the transit industry. Many transit providers describe Route Direction as one of opposite directions, East versus West, or inbound versus outbound. These opposites may be abstracted to be “first” and “second” direction. Alternatively, some transit providers will use the station name or city near the final destination of the route to signify the route direction (e.g., Babylon and New York City). Furthermore, some providers use codes for their destinations. In some cases, transit agencies use different codes for direction as well as indicating “bound” (in or out).

The SDP was designed to provide flexibility to transit providers so they could use any of these approaches and still meet the requirements of the SDP. The CDRM supports the ability to implement a “lookup” table for each route direction, namely the Route Direction element which is embedded in the Route element.

Requirements for Route Data Concept

The requirements associated with the Route Data Concept are listed in Table 4.3-1.

Table 4.3-1: Route Requirements

#	Category	Requirements
1	Uniqueness and identity	• A route has a schedule name and identifier as well as a public name and identifier. (Names and identifiers may be identical.) • A route is a collection of patterns and associated trips. Combined the patterns represent a branching set of paths over which a revenue vehicle traverses in up to two directions. • A route is valid for a specified schedule version. • A route has a designated validity period (or begin date) for all its associated records (see Associations) that compose the route. The validity period in most cases is the same as the schedule version activation/deactivation dates, and the validity period should fall within the schedule version period.
2	Route name	A route identifier/name may be different based on the user or type of route. • Schedulers identify a route by an identifier and name (e.g., M 0001, FIFTH-MADISON-PARK AV). • The public may recognize a route by a known number and name (e.g., M1, Fifth and Madison Avenues) • A route will include a description of its service provision, e.g., regular or express.
3	Route Type	The collection of service provision (trips and patterns) grouped by a route number may differ depending on the route type. Several types supported by NYCT include public route, curtain route (headsign), timetable, schedule route (group of routes that operate on the same corridor), statistical route (operator routes). • The SDP requires the most basic route type. In most cases the public and scheduler route service provision are identical. In which case in designating the scheduler's route number, the publicly known route number may be substituted. However there are exceptions to this rule when the public route number is used to collect multiple scheduler routes or in more complicated cases, when a scheduler's route description must be separated into multiple route numbers that the public understands. In the latter case, NYCT generates a Route 6191, which the public knows as two separate "curtain" routes, although they are scheduled together and displayed in a single timetable as the S61 and S91. • This may necessitate the ability to version routes by a schedule route number and a public route number. [A route grouping artifact can support this requirement.]

Table 4.3-1: Route Requirements

#	Category	Requirements
4	Associations	The route may be associated with key service and transit network descriptions including: • List of patterns categorized by route direction (or general direction of travel) • Stick map with designated stop/timepoints • Schedule Notes • Set of valid trips associated with each pattern • Information on route designation of accessibility (assumes access at all stops and vehicles) • General fare information • Connections with other routes
5	Route Hierarchy	• Service by Route Direction • Service by Day Type
6	Attributes	• Notes on service. This should only be used to support customers or provide information to schedulers. Other attributes, type descriptions and codes should be used prior to using a note. • Mode type

Conceptual Data Reference Model Description for Route

The Route requirements that are from Table 4.3-1 may be implemented in a CDRM as depicted in Figure 4-11. The data model description (as excerpted from the *Functional Requirements* document) follows.

> "A Route is a collection of patterns classified by their route direction in scheduled service with a common identifier. As shown, Route is an entity associated with patterns and related trips. One or more directed patterns may optionally be associated with a Route, and one Route categorizes one or more patterns through a route direction. One or more trips may optionally be associated with each pattern.
>
> "A Route contains attributes that support a transit customer. The Timetable_Header is a summary of published timepoints (or transit stops used as timing points) contained in a group of Patterns oriented in the same route direction, and is used to generate timetables. A Route is valid during its designated period (routeBeginDate to routeEndDate) that falls within a valid schedule version.
>
> "Similar to a Route, a Route Grouping is a collection of patterns, trips or both. A route grouping is assembled for a specific purpose other then a typical Route definition."

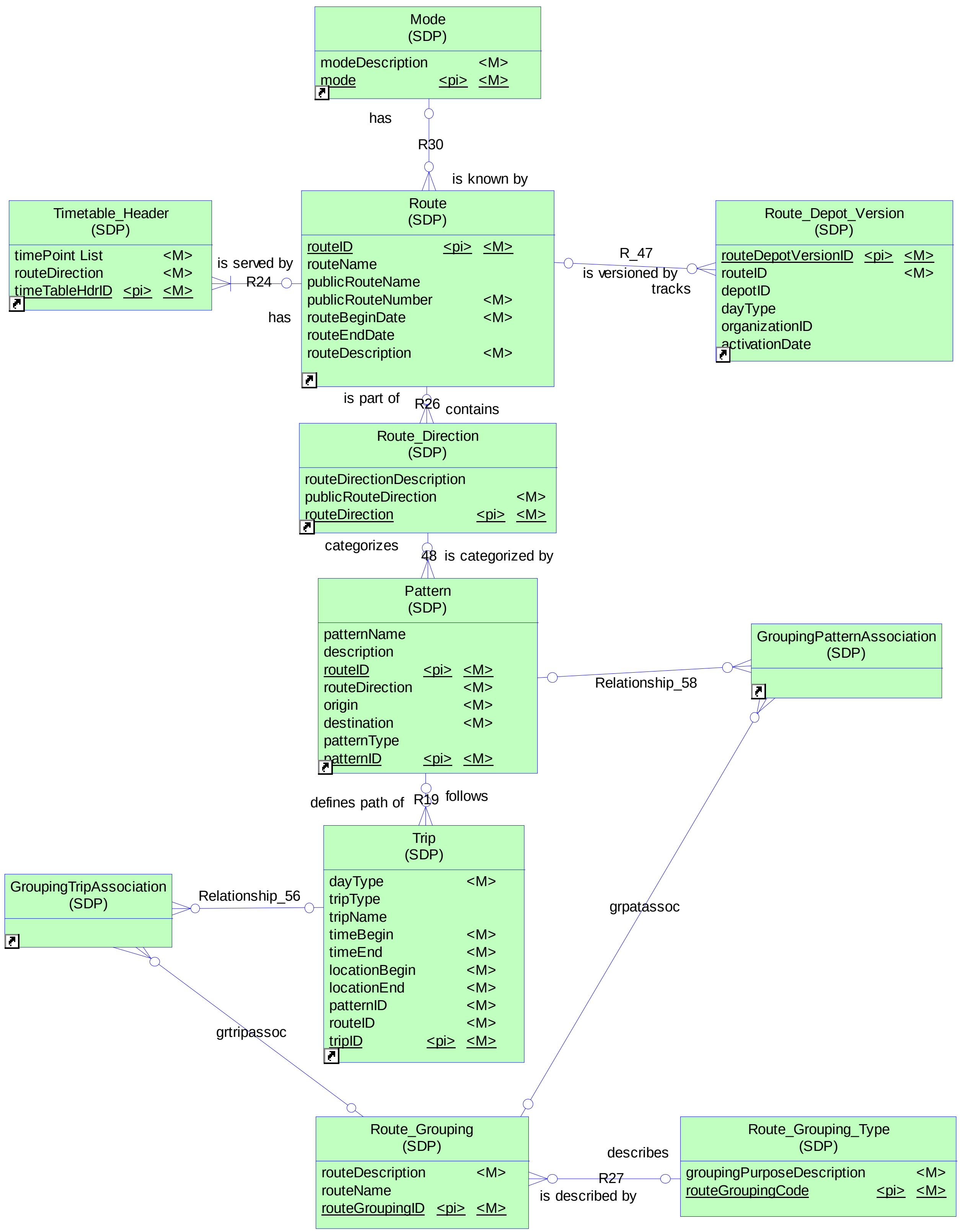


Figure 4-11: Route Conceptual Data Reference Model

Route Excerpt of XML Schema Model

The following rules and assumptions were used to implement the CDRM Route Data Concept as the Route element in the SDP XML Schema:

- The Mode entity is transformed to a code value, for example, use "CR" for commuter rail, "HR" for subway, "MB" for bus, and "FR" for ferry.
- Route Direction and Timetable Header entities are embedded as a list of routeDirectionList and timetableHdrList elements, respectively.
- Mandatory elements include routeID, mode, publicRouteName, publicRouteNumber.
- Attributes effectiveDate and endDate refer to the date when the record was placed, while the elements—routeBeginDate and routeEndDate—represent the dates when service begins and ends for this particular collection of patterns and trips.

Figure 4-12 depicts the Route element portion of the SDP XML Schema as it was derived from the CDRM.

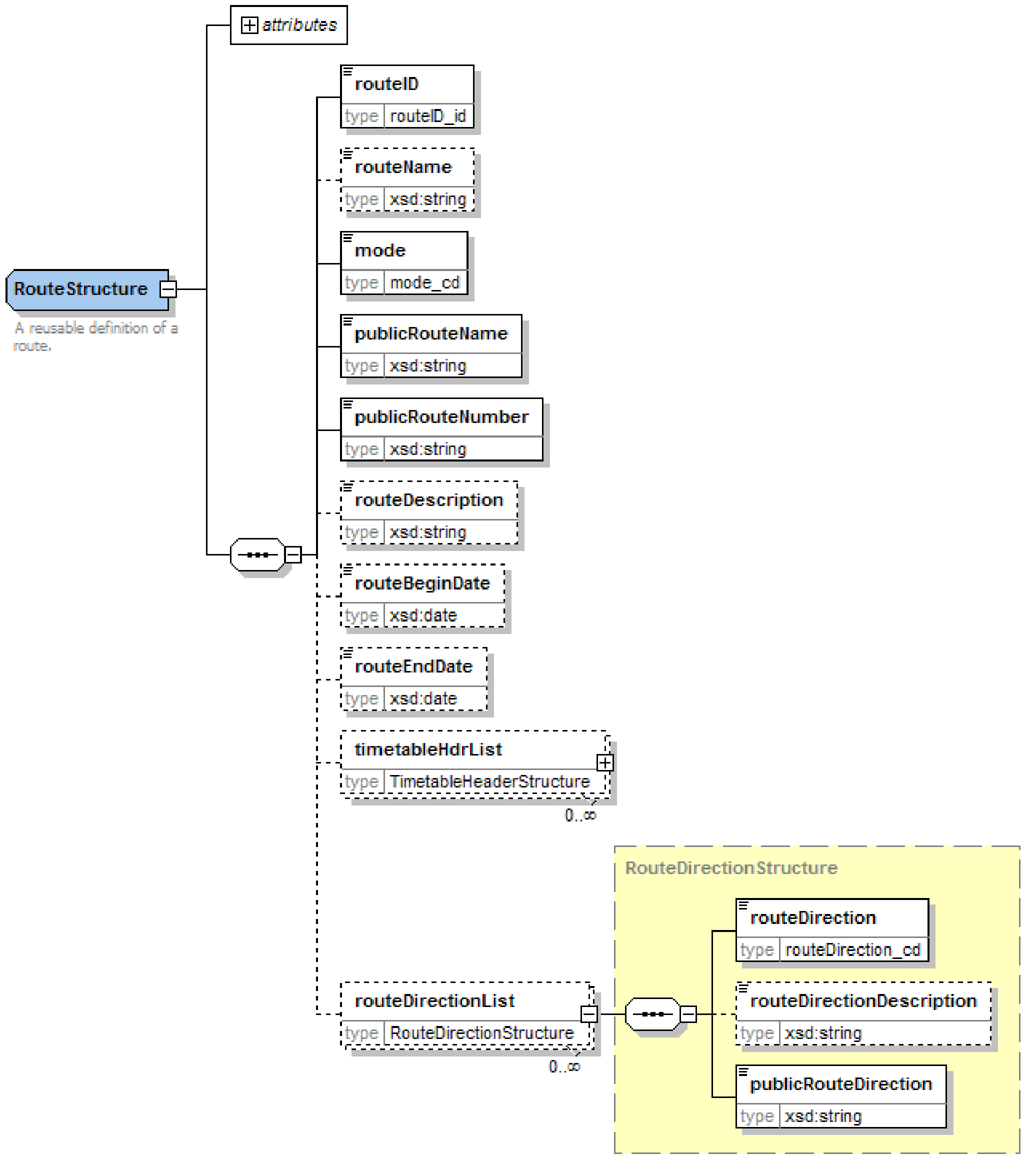


Figure 4-12: Route Element Fragment in SDP XML Schema

As listed in Table 4.3-1, the Route Data Concept governs the requirements for the SDP XML Schema elements in the Route element. In particular, the publicRouteName and publicRouteNumber are used by applications that display information to a customer, as such, these are mandatory elements.

The Route Direction data concept is directly implemented from the CDRM. Table 4.3-2 lists the Route Direction data elements from the CDRM requirements. The specific rules and assumptions include:

- routeDirection is the enumerated code value that is used to match the routeDirection element in the Pattern element. Typically, this is “first” or “second” direction. Care should be taken that the translation from native data follows a conventions such as:

North, West, inbound, circular (clockwise) are translated as "first", and South, East, outbound, (counter clockwise) are translated as "second."
- publicRouteDirection is the direction that a customer knows from a timetable. Because a route contains patterns that do not always contain the same end location, using destination information is not always the best practice.

The routeDirectionDescription field should be used by the schedule providers to support internal or specific application needs. For example, some agencies may use the routeDirectionDescription as a source for translating from their native route direction codes to TCIP SCH-RouteDirectionName. TRIPS123 uses the TCIP SCH-RouteDirectionName enumeration, which may not be represented using the "first"/"second" or in the publicRouteDirection fields. To support the mapping to TRIPS123, the content of the routeDirectioDescription may store the TCIP code value using the following notation TCIP= "north" (where enumerated type "north"—note spelling and capitalization) may be used to tag the TCIP value.

Table 4.3-2: Route Direction Element Description

Element Name	Definition	Examples
routeDirection	A unique identifier for a route direction. Codes: "first" or "second"	first
publicRouteDirection	A name known to the public as the direction of the route, e.g.., inbound, outbound.	North
routeDirectionDescription	Describes the direction as known to the scheduling group or other key stakeholder. Headsign information may be contained in this field.	1N17 HEMPSTEAD DIRECTION North

A best practice related to generating the routeDirectionList is to adopt and document a set of conventions used to translate native route direction codes or values to first or second routeDirection enumerated types. Conventions might be:
- "first"—N, W, NW, SW, Clockwise, Circular, Inbound
- "second"—S, E, SE, NE, Counter-Clockwise, Outbound

Detailed Data Descriptions and Guidance for Route and Route Direction

This section describes the format and guidance associated with Route and RouteDirection elements in the data concepts described above. The guidance for each element is consolidated into two tables—Table 4.3-3: Route Guidance and Table 4.3-4: Route Direction Guidance. Each table has the following column headings: Requirement status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML document deployment.

Table 4.3-3: Route Guidance

Required	Element Name	Type	Questions to Ask
M	routeID	routeID_id UNIQUE	What designation for route is unique within your data source?
O	routeName	String	What field is used to name the route?
M	mode	Mode_cd	List mode code from mode_cd; e.g., MB for bus, CR for commuter rail, HR for subway (heavy rail), FR for ferry, etc.
M	publicRouteName	string	What field represents the route name recognized by the public?
M	publicRouteNumber	string	What field represents the route number recognized by the public?
O	routeDescription	string	Is there a field that describes the route? This field is not needed if it is redundant with publicRouteName. The field may be used to describe the physical path of the route, or if it contains abbreviations for other internal or machine-to-machine data transfer.
O	routeBeginDate	date	Do all routes start and end on the same dates as the schedule version? y/n • If no: are there fields that indicate the dates a route begins and ends?
O	routeEndDate	date	If no end date is present in a native route record, then use one of these default values (1) schedule end date or (2) default end date 9999-12-31
O	timetableHdrList	TimetableHdr	Is there a set of fields that is used for a published timetable header? The complete TimetableHeader record.
O	routeDirectionList	RouteDirection	First and second directions are used for route direction. This record is used to customize and provide a public name for the specific route route-direction. Complete RouteDirection record for the number of route directions that are listed by the patterns in this route.
O	@effectiveDate		
O	@endDate		

Table 4.3-4: Route Direction Guidance

Required	Element Name	Type	Questions to Ask
M	routeDirection	routeDirection_cd	Enumerated index used to match routeDirection in Pattern element.
O	routeDirectionDescription	string	The name or code used in the native data. Content is determined by the user, and may be used to support downstream applications.
M	publicRouteDirection	string	The route direction name known and viewed by the public.

Examples Using Route

Long Island Bus Route N1 Jamaica-Elmont-Hewlett

This example shows how the SDP XML Schema allows for the simultaneous use of two different route designators for the same route, and the use of two different types of codes to indicate route direction.

The Long Island Bus uses different route designators for their identifier and public number or name. Furthermore, they may use different conventions when documenting the route number, for example N01 versus N1. Their native data includes a data construct called "bound" which is assigned a zero (0) or one (1). Their direction code refers to compass directions (North, South, East and West).

```
<Route>
  <routeID>3210</routeID>
  <routeName>N01</routeName>
  <mode>MB</mode>
  <publicRouteName>N1 Jamaica-Elmont -Hewlett</publicRouteName>
  <publicRouteNumber>N01</publicRouteNumber>
  <routeDescription>N1 Jamaica-Elmont -Hewlett</routeDescription>
  <routeDirectionList>
      <routeDirection>first</routeDirection>
      <routeDirectionDescription>bound=0 direction=3 tcip=1</routeDirectionDescription>
      <publicRouteDirection>North</publicRouteDirection>
  </routeDirectionList>
  <routeDirectionList>
      <routeDirection>second</routeDirection>
      <routeDirectionDescription>bound=1 direction=4 tcip=2</routeDirectionDescription>
      <publicRouteDirection>South</publicRouteDirection>
  </routeDirectionList>
</Route>
```

New York City Transit Rapid Rail (RTIF)

This example shows how the SDP XML Schema allows for the exclusion of optional elements, such as when agency data are not available for the routeDescription element.

The native data set from which the MTA NYCT Line 2 SDP XML Document was derived does not have more information for the public about the route beyond a simple publicRouteName and publicRouteNumber. Therefore, there is limited information to insert in those fields and only redundant information to include in the routeName or routeDescription fields; since these latter two elements are optional, we chose to only include the routeName.

The native data set only included North/South directional pairs. As a result, in the data conversion process to the SDP XML Document, the recommended convention for coding “first” and “second” was used (e.g., first for North and second for South).

```xml
<Route>
  <routeID>2</routeID>
  <routeName>Line 2</routeName>
  <mode>HR</mode>
  <publicRouteName>Line 2</publicRouteName>
  <publicRouteNumber>2</publicRouteNumber>
  <routeDirectionList>
    <routeDirection>first</routeDirection>
    <routeDirectionDescription>N</routeDirectionDescription>
    <publicRouteDirection>North</publicRouteDirection>
  </routeDirectionList>
  <routeDirectionList>
    <routeDirection>second</routeDirection>
    <routeDirectionDescription>S</routeDirectionDescription>
    <publicRouteDirection>South</publicRouteDirection>
  </routeDirectionList>
</Route>
```

Section 4.4: Organization Unit and Depot and Related Data Concepts

In This Section

- Explore the Organization Unit and Depot Data Concepts pertaining to the Agency Registration Branch.
- Learn when and how to apply the elements in the Organization Unit and Depot data concepts.

Organization Unit Definition
An administrative entity or agent (e.g., contractor) of a transit agency. The unit may be a department or service area. An organizational unit may be part of a larger organizational unit, for example, a contracting service may be overseen by the agency's operations department.

Depot Definition
A storage facility operated by a transit agency or its agent where transit vehicles park overnight and are maintained.

Purpose of the Organization Unit and Depot in the SDP

The Organization Unit and Depot data concepts may optionally be included in a SDP XML Document submittal. The Organizational Unit is included in the SDP CDRM to reference the responsible party that generates or submits a schedule data set. Sometimes, a large transit provider may generate schedules by Organizational Unit. For example, bus and rail operations generate schedules at different times of the year, and so may have different schedule versions.

The Depot data concept is included because a provider with a large bus fleet or with varied work rules by garage may generate schedules by routes and trips originating from a specific Depot. The Organizational Unit and Depot data concepts provide descriptions of the group for which generated a specific schedule data set. To date, formal descriptions of depot and organization unit are not needed by current downstream applications, however, the concepts are included in order to be comprehensive and meet potential future needs.

Please note that these SDP data concepts are not used to describe **who** submitted the SDP XML Document; submitter contact information is incorporated in the Metadata information. (See Part 3 SDP Guidance for more information on Metadata.) Organizational Unit and Depot elements are used to describe their association to the schedule contained in the SDP XML Document.

Requirements for Organization Unit and Depot Data Concepts

The requirements associated with the Depot and Organization Unit Data Concepts are listed in Tables 4.4-1 and 4.4-2, respectively.

Table 4.4-1: Depot Requirements

#	Category	Requirements
1	Uniqueness and identity	• A depot is assigned a unique identifier. The identifier may be a part of a collection of identifier just for the depot or garage, or it may be a facility identifier that is a transit facility identifier.
2	Depot Name	• A depot has a name, usually informal and formal names.
3	Depot Type	• The primary purpose of the depot may be a vehicle garage or maintenance center, operator base or other type of vehicle storage center. Furthermore, the operators and vehicles in the garage may provide service in one or more modes, however, there is typically a primary mode associated with the depot. The primary mode is used to designate the mode.
4	Associations	• A depot is associated with a location and may be associated with transit facility.
5	Attributes	• The vehicles and operators assigned to a depot usually cover a specific service area. The service area may be described with words or may be designated by geographic features that circumscribe their boundaries.

Table 4.4-2: Organization Unit Requirements

#	Category	Requirements
1	Uniqueness and identity	• An organization unit is assigned a unique identifier. This identifier will be used to index routes by organization.
2	Organization Unit Name or Description	• An organization unit may be named or described. Some agencies procure contracting agents for their service provision. The agents develop schedules and may be responsible for submitting the data sets. Information about the responsible party generating the schedule data set may be needed by the Agency.

Conceptual Data Reference Model Description for Organization Unit and Depot

The above requirements may be described in the CDRM for Organization Unit and Depot depicted in Figure 4-13. The CDRM model is described by the following:

> All or part of an Agency's schedule, the Route Depot Version, may be associated with a Depot or an Organization Unit. A Depot consists of one or more modes and its service typically covers a specific geographic area. The Organization Unit belongs to an Agency.

In addition to Organization Unit and Depot, this CDRM identifies two other data concepts that are nested in the SDP XML Schema (and are described as entities in the CDRM). The Mode and Services Area are defined below:

> Mode is "[a] transit service type category characterized by specific right-of-way, technological and operational features." [NTCIP 1401:2000] The values used are based

on the U.S. DOT National Transit Database [http://www.ntdprogram.gov/ntdprogram/Glossary.htm] values for mode.

Service Area is "[a] geographic area where transit service is provided." The area is described as a polygon circumscribed by a series of points. The Geographic Markup Language (GML) is used to define a service area feature. GML allows a polygon type to be instantiated as one of several geometries (e.g., point, polygon, multipolygon, etc.). GML is a reference to ISO 19101 standard.

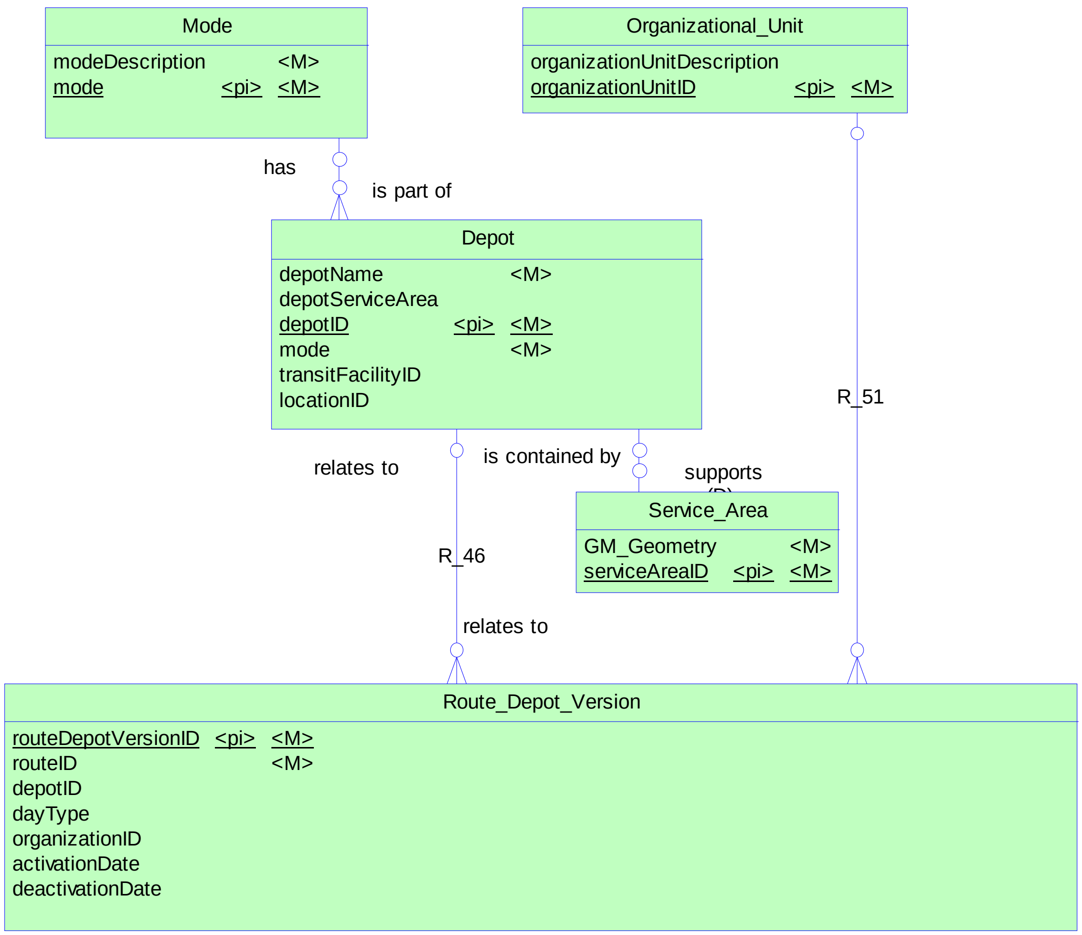


Figure 4-13: Conceptual Data Reference Model for Organization Unit and Depot

XML Schema Descriptions for Organization Unit and Depot

In implementing the CDRM into the XML Schema as shown in Figure 4-14, the following rules were applied.

- In the Depot data concept, the rules included:
  - The Mode data concept is embedded as an element that cites a value from an enumerated type.
  - locationID refers to a Transit Gazetteer branch Location identifier.
  - transitFacilityID refers to a Transit Facility branch Transit Facility identifier.
  - depotServiceArea may be instantiated as a GML polygon (see example below).
- No special rules were imposed on Organization Unit.

Depot and OrganizationUnit are optional child elements of Agency Registration. Zero or more of these elements may be included in the SDP XML Document. Both elements may be assigned effectiveDate and endDate as attributes.

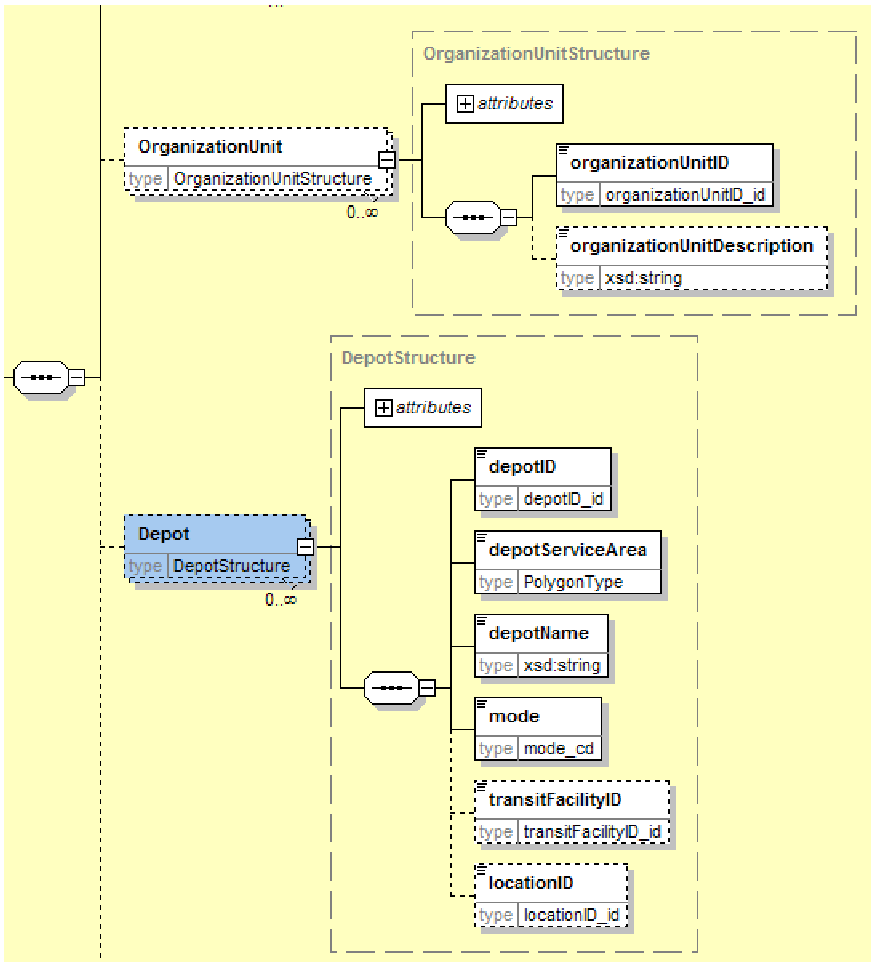


Figure 4-14: SDP XML Schema Excerpt of Organization Unit and Depot

Detailed Data Formats and Guidance for Organizational Unit and Depot

This section describes the description and guidance associated with OrganizationalUnit and Depot elements in the data concepts described above. Table 4.4-3 provides guidance on Organizational Unit and Table 4.4-4 provides guidance on Depot. The guidance for each element is consolidated into a table with the following column headings: Requirement status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML document deployment.

Table 4.4-3: Organization Unit Guidance

Required	Element Name	Type	Questions to Ask
M	organizationUnitID	organizationUnitID_id UNIQUE	Is there a field that designates the organization? Yes: <<field name>> No: assign number (e.g., “1”)
O	organizationUnitDescription	string	Is there a description, name or value associated with the organizational unit?
O	effectiveDate	date	[attribute]
O	endDate	date	[attribute]

Table 4.4-4: Depot Guidance

Required	Element Name	Type	Questions to Ask
M	depotID	depotID_id UNIQUE	Is there a field that designates the depot? Yes: <<field name>> No: assign number (e.g., “1”)
O	depotServiceArea	gml:PolygonType	This is a set of one or more sets of latitude and longitude that outline the service area. The schema for gml:PolygonType is included in GML_Geometry.xsd
M	depotName	string	Is there a name for the depot?
O	mode	mode_cd	Is there a field for the mode serviced by the depot? Are all the routes in this file of one mode? (then use a default value)
O	transitFacilityID	transitFacilityID_id	Is the depot associated with a transit facility?
O	locationID	locationID_id	Are there fields indicating the location of the depot? [trigger Location Table input]
O	effectiveDate	date	[attribute]
O	endDate	date	[attribute]

Usage and Examples of Organization Unit and Depot

Organization Unit

Within the scope of the SDP, the Organizational Unit is used as an index to an internal unit or mechanism that may initiate or produce a schedule. Examples of potential Organizational Units include:

- New Jersey Transit produces two schedule sets, one for rail and the other for bus and light rail. The bus and light rail schedule version may be aggregated by using the Organizational Unit entity. The Rail schedule version may still be associated solely with Mode.
- Bee-line (Westchester County) contractors each produce a detailed schedule. The Organizational_Unit may represent each contractor that produces a schedule.

The Organizational Unit entity may be used as an internal agency mechanism to classify schedule versions for roles other than agency, route, depot or route/depot. The association of organizationUnitID and its description should be maintained by the submitting agency, and as such, is included in the SDP as an internal reference; however, the element will not be used by the TSDEA at this point in time.

When the agency issues a single schedule for all organizational units, such as Long Island Bus, a direct relationship is established between the Schedule Version and Agency (and the organizational unit is not needed).

The example below shows how an Organization Unit would appear if the Metro-North Railroad bus operations issued their bus schedules for the Hudson Rail Link.

```
<AgencyRegistration>
…
        <OrganizationUnit>
                <organizationUnitID>1</organizationUnitID>
                <organizationUnitDescription>Hudson Rail Link </organizationUnitDescription>
        </OrganizationUnit>
…
</AgencyRegistration>
```

DEPOT

The example SDP XML Document fragment used as an example below is translated from the NYCT schedule data files based on their internal exchange format. The NYCT Bus files are generated and versioned by Depot. The Depot is assigned a location identifier. The depotID and transitFacilityID in this case are identical.

```
<AgencyRegistration>
…
        <Depot>
                <depotID>KB</depotID>
                <depotName>Kingsbridge</depotName>
                <mode>MB</mode>
                <transitFacilityID>KB</transitFacilityID>
                <locationID>3eac</locationID>
        </Depot>
…
</AgencyRegistration>
```

Service Area (an nested element of Depot)

The example of a GML Polygon, “a closed region of space”, is of a box with four points. The polygon, by definition may contain as many points as needed to define the region. The major requirement is that the points are ordered and the first and last points are the same. Since the SDP specifies the use of WGS ’84, the coordinates should use the same datum. The GML polygon example is defined as follows:

[from http://geoweb.blog.com/278606/ and cited in the GML 1.0 and 2.0 standards]

```
<gml:Polygon>
    <gml:outerBoundaryIs>
        <gml:LinearRing>
            <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
        </gml:LinearRing>
    </gml:outerBoundaryIs>
</gml:Polygon>
```

Chapter 5: Service and Related Data Concepts

In This Chapter

- Understand the requirements related to the Service Branch.
- Discover how Service branch and related data concepts are used.
- Learn how to apply the elements in the Service branch.
- Understand how to convert Standard Daily Time to Schedule Time for the SDP.

Service

The service branch describes the provision of (revenue) service. It includes the following elements:

- Trip
- Note
- Block (bus assignment)
- Event Connection (discussed in Chapter 10)

Purpose of a Service Branch Model

The Service Branch of the SDP XML Schema provides needed information on the transit service provided by an agency or organization unit of an agency. The Service Branch elements include information on trips and trip times, scheduling notes, bus assignment schedules (i.e., Block) and coordinated transfers (i.e., Event Connection). The Block and Event Connection requirements and guidance are described in Chapter 10: Advanced Topics.

Conceptual Data Reference Model (CDRM) Description for Service

The CDRM, shown in Figure 5-1, describes the data concepts that are incorporated in Service and how they are related. Trip is the primary building block of Service. The key portions of the Service CDRM are described as follows:

- A Trip is a one way scheduled movement of a transit vehicle between starting and ending locations (typically timepoints) which operates on certain days. A Trip is classified as revenue, non-revenue, and sometimes by agency defined types. Each Trip is an instance of a Pattern.
- A Trip may be associated with zero or more Notes. A Note may further describe the trip, for example, such as: “Connecting Service” or “Friday Only.”
- The Trip is demarcated by an ordered sequence of Trip Time event times and locations. A trip time location may be a timepoint, transit stop, or both, as well as a location used to trigger an operational event.
- A Trip is composed of two or more ordered 'times' (tripTime) of Trip Time. Each tripTime in the sequence becomes a temporal event of the Trip describing events that occur at that location. A TRIP TIME may occur at a specific location more than once in a single TRIP, however, each occurrence is a unique temporal event (tripTime). (For example, a revenue vehicle may loop to the same stop more than once during a single trip. Another example may involve a train that arrives at a station at 10:30 a.m. and departs the station at 10:40 a.m. These may be described as two separate events at the same location. Each is described as a separate Trip Time.) Trip Time includes:
  - tripTime (mandatory) in the form of a signed integer that represents seconds past midnight. (The value may exceed a 24 hour clock.)
  - locationID (mandatory) of the event (should match a point on the associated Pattern).
  - timeType (mandatory) which describes whether this is an arrival, departure, passing, begin trip or end trip time.
  - platformNo (optional) for trains that include a boarding area in the Trip Time element.

  - seqNo (optional) or sequence number that defines the order in which the trip times occur. Alternatively, the tripTime, since it is an integer, may be used to sort the Trip Times.
- A Trip Time may be composed of zero or more Trip Event Types.
- The Trip Event Type is a classification of the event that occurs at the Trip Time. The Trip Event Type includes a designation that the event is included in the timetable header, short turn occurs, and alighting, boarding or both. Additional values and descriptions may be inserted as needed.

Additionally, a Trip Time may be associated with a scheduled connection to a different Trip. The Event Connection should include the time of the arriving Trip Time event and the departing Trip Time event. (A more detailed discussion of the coordinated transfers is discussed in Chapter 10: Advanced Topics. The discussion describes the associations among Event Connection, Connection Segment and Transfer Cluster.)

A Trip Time may be associated with zero or more Notes. A Note may further describe the event, for example, Metro North Railroad includes notes such as: "Discharge may depart 5 minutes early," or "Connecting Service."

A Note Library exists which is composed of Note_Entry(ies). Each entry contains a note identifier (noteID) and noteText that is shown to the public.

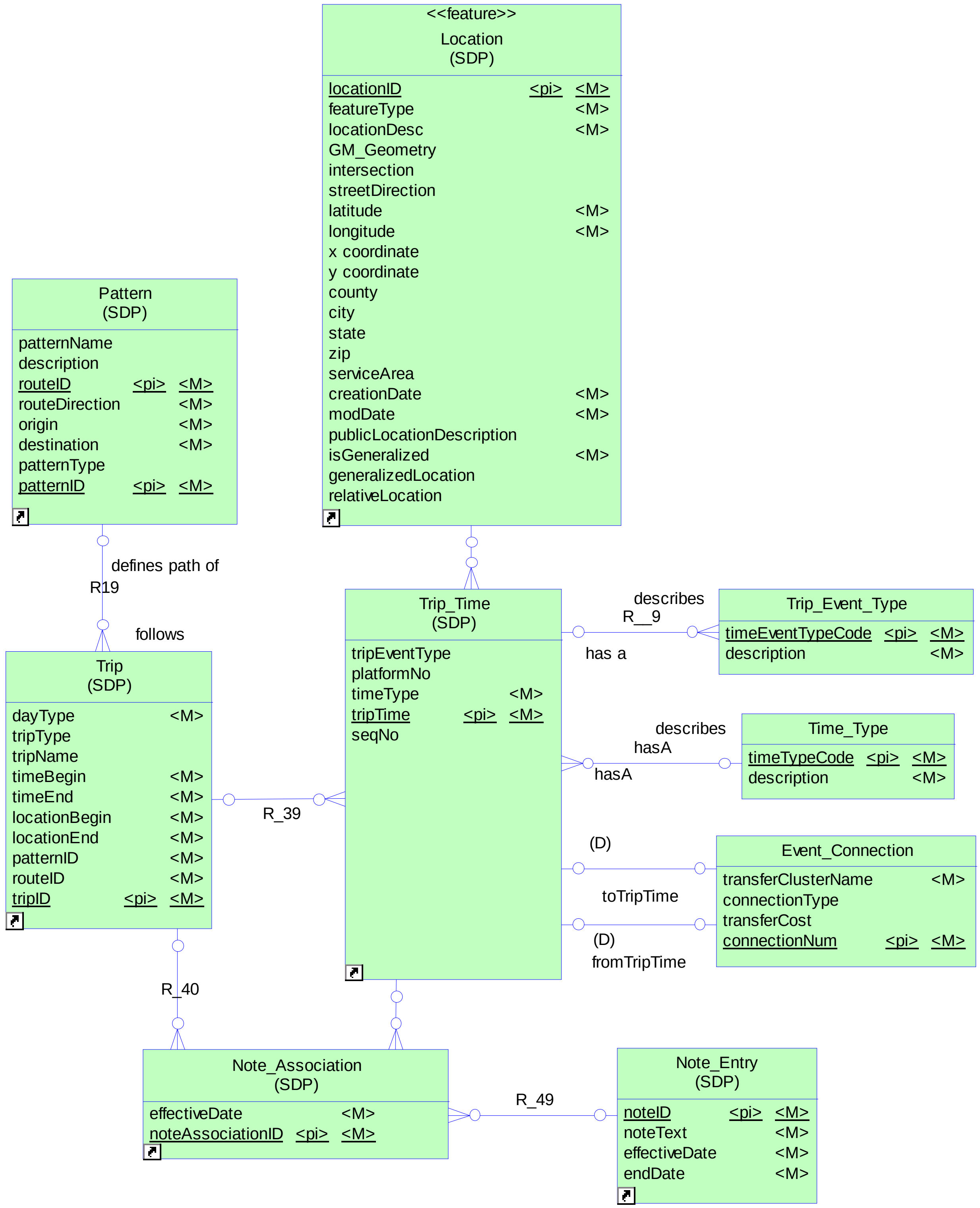


Figure 5-1: Service Data Model (Without Block)

Each data concept depicted in the Service CDRM will be discussed in more detail later in this chapter.

SDP XML Schema Description for the Service Branch

The rules applied to implement the CDRM as an XML Schema "flattens" the relationship among the entities in the Service Provision layer. Figure 5-2 depicts the top level of the Service branch from the SDP XML Schema.

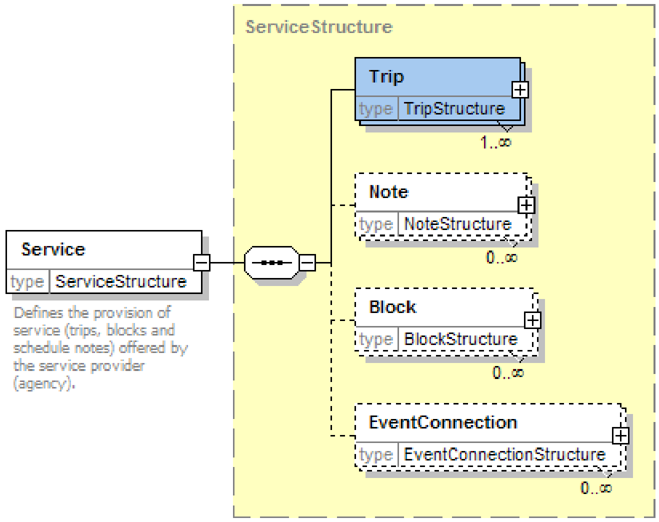


Figure 5-2: Service Model Implemented in the SDP XML Schema

Detailed descriptions of embedded elements in the Trip and Note elements may be found in Sections 5.1 and 5.2, respectively. (The Block and Event Connection requirements are described in Chapter 10: Advanced Topics.)

Section 5.1: Trip and Related Data Concepts

Purpose of the Trip Data Concept

Trip and its related entities, Note_Association, Trip_Time and Pattern, describe key elements of the transit service referenced by a schedule. Agencies typically have both revenue and non-revenue trips. Currently, non-revenue trips are not used by regional downstream applications, but they may be needed by internal applications that use the SDP format.

Requirements for the Trip Data Concept

The requirements that drive the Trip Data Concept are described in Table 5.1-1.

Table 5.1-1: Trip Requirements

#	Category	Requirements
1	Unique/Identity	• A trip is unique for a schedule version developed by a Transit Agency. • A trip is the temporal component of a pattern of a route in a given direction on a given type of day. • Each trip is fully defined by the set of scheduled times that a vehicle passes (arrives at or departs from a point) along an associated pattern. Specifically, a trip is composed of two or more ordered times. The times are typically described with respect to a schedule day (e.g., 24 or 36 hour day). • *Identity:* A unique identifier may be used to define a trip. The identifier should not be reused within a schedule version. Although not recommended, a trip may be identified by the first "trip time," i.e. passing time at the first timepoint. • A trip may be described as a temporal instance of a pattern. • Elements (scheduled trip times) of a trip correspond to selected points along a physical path described by a pattern. • Rail transit may identify a customer's "service trip" by the primary train number or name. (See Appendix A for more information on Special Considerations for Rail Transit).
2	Composition	• A trip is described by a starting location and departure time, and by an ending location and arrival time. The starting and ending locations should correspond to the origin and destination locations of the referenced pattern. • A trip may only operate on specific, non-overlapping days (e.g., weekday, Monday only, holiday) called Day Types.

Table 5.1-1: Trip Requirements

#	Category	Requirements
3	Trip Collections	• A set of abridged trip descriptions, aggregated by route direction and day type may be published for customer use as a timetable. • The set of trips, each associated with a pattern, and each pattern belonging to a route in a given direction, constitutes the scheduled service for a route. • A set of trips from one or more routes may be scheduled together in order to operate frequent service along a corridor. These are sometimes called lines or route groupings. • A set of trips may be chained together to describe the vehicle assignment or work for a day (block). The ordered set of trips may be characterized as revenue and non-revenue. The first and last trips (usually non-revenue) typically detail the vehicle's pull out from, and pull in to the garage or depot.
4	Exceptions	• As an operational exception, a trip may follow only part of a pattern if it was developed as a "short turn" trip. The point at which it turns around should be described as a ***time event type*** in the trip time description.
5	Subtype	• A trip may be revenue generating. • A trip may be non-revenue generating such as a deadhead, pull in or pull out. • A trip may be characterized by the type of service provided including regular, express, limited, scenic, etc.
6	Attributes	• Schedulers may sometimes attach one or more notes to a trip or trip time to convey information about the scheduled service. • Other attributes may sometimes be associated with a trip such as amenities that should be offered by the operating vehicle (e.g., bicycles rack/allowed), accessibility, mobile WiFi, etc.

Conceptual Data Reference Data Model Description

The Trip requirements are modeled in the CDRM as previously shown in Figure 5-2. The Trip and associated entities are used by Route, Route Grouping and Block as components of the transit service description.

SDP XML Schema Model Description for TRIP and TRIP TIME

In the course of implementing the SDP XML Schema for Trip from the CDRM, a number of translations were used. The XML Schema excerpt of the Trip data concept is illustrated in Figure 5-3. (Note: Figure 5-4 expands tripTimeList to show the elements of the embedded Trip Time element.)

The rules and assumptions associated with the Trip element include:

- Implemented *type* entities as enumerated type structures embedded in Trip element. The entities changed include: tripType, dayType.
- noteList embeds the set of related noteIDs into the Trip element.
- timeBegin and timeEnd should correspond to the first and last tripTime child element of tripTimeList elements.

- Likewise, locationBegin and locationEnd should correspond to the first and last locationID child element of the tripTimeList elements.
- timeBegin and timeEnd use schedule time, that is the number of seconds past midnight (a signed integer).

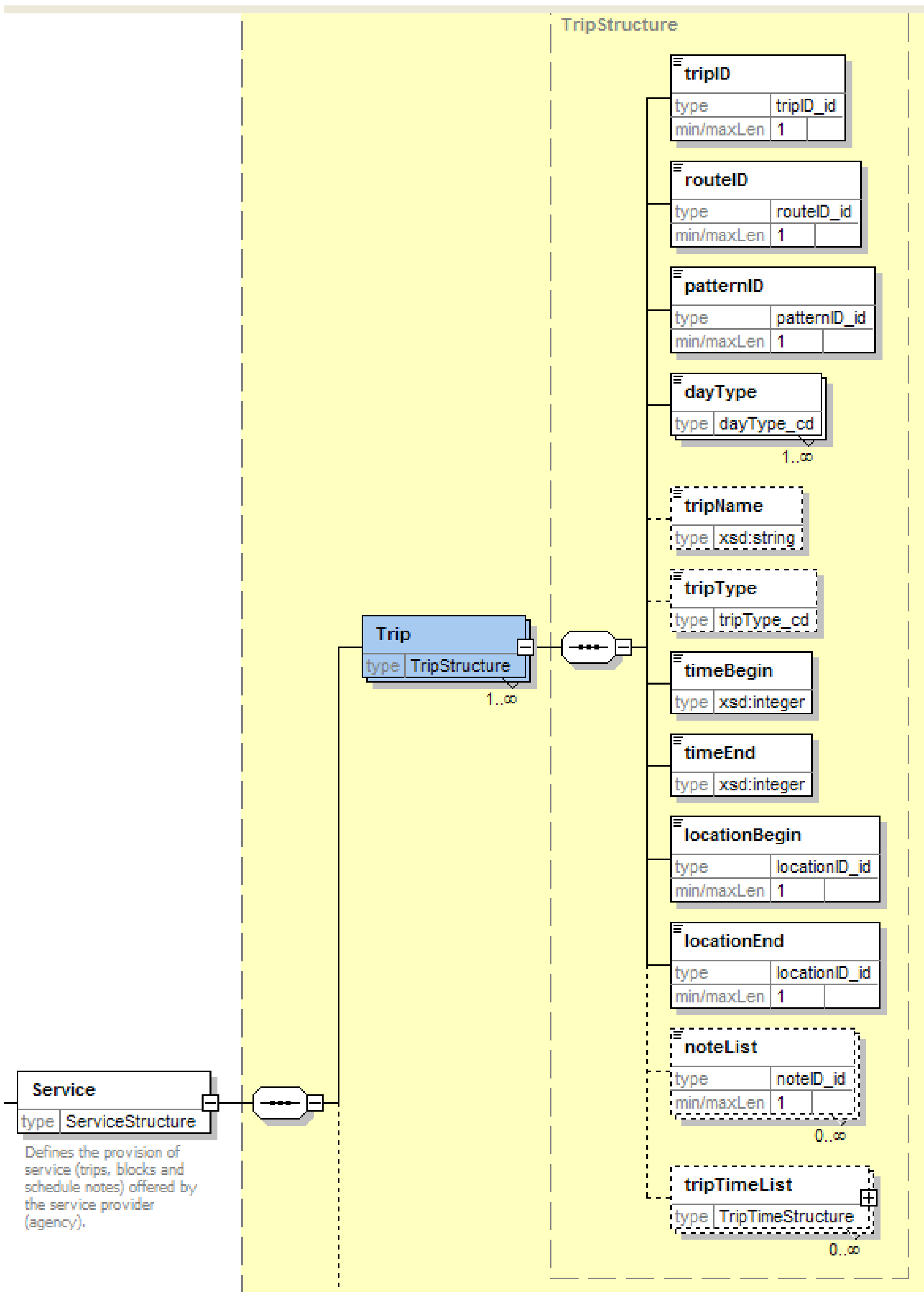


Figure 5-3: SDP XML Schema Excerpt of Trip

Figure 5-4 shows the SDP XML Schema for TripTime (this embedded element is tagged with tripTimeList in the Trip element). The rules and assumptions associated with the Trip Time element are similar to Trip. They include:

- Implemented *type* entities as enumerated type structures embedded in Trip Time element. The entities changed include: tripEventType, timeType.
- notes embeds the set of related noteIDs into the Trip Time element.
- tripTime, seconds past midnight, uses a signed integer, and thus may be used to sort the elements within the Trip, however, seqNo is added for those applications that prefer a set of sequential non-negative integers.
- Some agencies may wish to relate this element to a specific stop. The platformNo is equivalent to a stopID, therefore, it may be considered a stop.
- A rail operator may also associate an event with the trackNo. See the discussion on extensions to the SDP XML Schema (Appendix A).

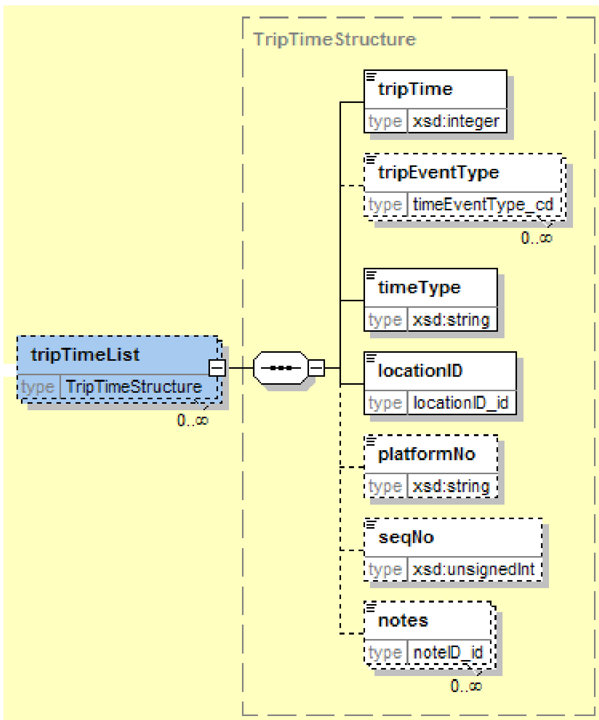


Figure 5-4: SDP XML Schema Excerpt of Trip Time

Detailed Data Formats and Guidance

This section describes the format and guidance associated with Trip in Table 5.1-2 and with TripTime in Table 5.1-3. The guidance is consolidated into tables that include a list of baseline requirements (M for mandatory and O for optional), the element name, the data type and guidance related to element. The first column of each table designates the baseline requirements based on the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet its data needs. The element name corresponds to the related conceptual data model entities and attributes. The type may refer to a native XML type, or declared type in the XML schema. The Guidance column is called

"Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

Table 5.1-2: Trip Guidance

Required	Element Name	Type	Questions to Ask
M	tripID	tripID_id UNIQUE	What field uniquely designates the trip? If there is no field, can you generate a unique set of identifiers per Route or across the schedule version data set?
M	routeID	routeID_id	How is a trip associated with a route?
M	patternID	patternID_id	Is a trip associated with a pattern? How?
M	dayType	dayType_cd [1..*]	How is day type defined and assigned to trips? (See Chapter 9 guidance on assigning day types). Notice that many day type codes may be used so that the trip definition may be used as a building block to develop sets of trips that may be used for different types of service. A special, locally defined day type may be defined using the DayType element in the AgencyRegistration branch.
O	tripType	tripType_cd	What field corresponds to tripType? Only revenue trips are needed for regional applications.
O	tripName	string	This field may be used by a named trip. Many rail operators may name a "train," for example, the field may be used as the primary train number for a service trip.
M	timeBegin	time	Time trip begins (timeBegin should correspond to first TripTime.tripTime value).
M	timeEnd	time	Time trip ends (last timeEnd should correspond to last TripTime.tripTime value).
M	locationBegin	locationID_id	Location begins (locationBegin should correspond to first TripTime.locationID value).
M	locationEnd	locationID_id	Location ends (locationEnd should correspond to last TripTime.locationID value).
O	noteList	list of noteID_id*	If there are notes, then complete the Note library and assign appropriate noteID to this list.
M	tripTimeList	ordered list of TripTime*	This is the list of trip times in the trip. Typically this is stored in sequential order, first time to last time of the trip.

Table 5.1-3: Trip Time Guidance

Required	Element Name	Type	Questions to Ask
M	tripTime	integer	The trip time is defined as an integer representing seconds past midnight. See Algorithm 1 below for technique to translate from clock time to seconds past midnight.
O	tripEventType	tripEventType [0..*]	Are there points or times along the trip that are designated for events (e.g., fare set change, headsign change, interior sign/annunciator message)? If true, then check code list for all the events represented in your data. This field may be listed more than once.
M	timeType	timeType_cd	Is the type of timing described in the table: arrival, departure, passing? Typically, if the tripTime is the first, then the default field is departing. If the field is the last, then the default is arrival. The default for all other occurrences depends on scheduling policy.
M	locationID	locationID_id	This locationID corresponds to the point along the trip where the time occurs. It may be any feature where time is measured.
O	platformNo	string	The platform where boarding or alighting occurs; check if mode is a Ferry or rail (commuter or subway). If true: How is the platform associated with the train or ferry's arrival at a station?
O	seqNo	integer	The order of this tripTime in the trip sequence. This field is not mandatory since the tripTime may be used to order the events.
O	notes	list of noteID_id	If there are notes, then complete the Note library and assign appropriate noteID to this list.

Usage and Examples of Trip and Trip Time Elements

Similar to a database, the SDP XML Schema allows an unlimited number of TRIP elements to be inserted between the Service tags. Additionally, an unlimited number of tripTimeList (TripTime) elements may also be embedded in each Trip element. Furthermore, the example below shows that optional elements may appear in one child element, yet not be present in another. This may be seen with the child element noteList in Trip, and again in tripTimeList with the tripEventType and notes child elements.

[Note: noteList and notes content were inserted to illustrate how the note association concepts work. They were not part of the native data.]

```xml
<Service>
  <Trip>
    <tripID>1</tripID>
    <routeID>3210</routeID>
    <patternID>119</patternID>
    <dayType>sat</dayType>
    <tripType>revenue</tripType>
```

```
        <timeBegin>22380</timeBegin>
        <timeEnd>22980</timeEnd>
        <locationBegin>7</locationBegin>
        <locationEnd>4918</locationEnd>
        <noteList>Express</noteList>
        <noteList>Friday Only</noteList>
        <tripTimeList>
          <tripTime>22380</tripTime>
          <tripEventType>ChgHeadsign</tripEventType>
          <timeType>beginTrip</timeType>
          <locationID>7</locationID>
          <seqNo>1</seqNo>
        </tripTimeList>
        <tripTimeList>
          <tripTime>22980</tripTime>
          <timeType>passing</timeType>
          <locationID>4918</locationID>
          <seqNo>2</seqNo>
          <notes>Alighting Only</notes>
        </tripTimeList>
      </Trip>
  <!-- more trips here -->
    </Service>
```

Converting Standard Daily Time to Schedule Time

Different transit agencies in the United States use different methods for describing time when a transit trip operates over midnight, existing, in part, on two different days. Standard daily time may not be used for some systems and reports. Instead a day may be extended from 24 hours to some other number of hours, such as 30 hours in “schedule time.” Also, some transit agencies may use either a 12 hour or a 24 hour clock for “schedule time.”

This section describes an algorithm to convert standard time, i.e., hh:mm:ssss to Schedule Time. The schedule time was converted from minutes past midnight to seconds past midnight. To convert from standard time to seconds-past-midnight, use the following algorithm:

Algorithm for a 24 hour clock

tripTime = hh*3600+mm*60+ss

Algorithm for a 12 hour clock using a.m. and p.m.

If a 12 hour clock is used, then PM implies 12 hours from midnight, so add 12*3600 to the tripTime.

Trip time (also timeBegin/timeEnd) is a signed integer to represent times that occur prior to midnight (when a trip begins before midnight but is part of the next day’s schedule, e.g., New Year’s Day). Trip time may extend past 24 hours when a trip extends into the next day (after midnight) but belongs to the previous day, e.g., New Year’s Eve day.

Section 5.2: Note and Note Association Data Concepts

Purpose of a Note in a Schedule

Notes are a key source of information on scheduled trips and trip times. A note may be targeted for the public, an operator, scheduler or other user of the schedule. The SDP focuses on those notes that are used by the public. An agency may not have notes or may not choose to include any notes in its schedule, therefore the Note data concept is optional.

Some examples of possible Notes are shown below.

- Indicates specially-designated weekend train allowing more than the regular eight bikes per train limit.
- Change at Jamaica. The track of your connecting train will be announced.
- Stops only to discharge (receive) customers.
- These buses continue to Walt Whitman Mall.
- Exit from first two cars.
- Trip operates via Union City upon request at time of boarding.
- Trip operates express via I-280 between Newark and Orange. Westbound trips to Livingston Mall operate to Park-Ride upon request.

Requirements for Note Data Concept

The requirements related to describing and populating a note is described Table 5.2-1.

Table 5.2-1: Note Requirements

#	Category	Requirements
1	Purpose and Identity	• The noteID must be a unique identifier within a schedule version. • A noteID may be defined as an alphanumeric character. • The noteText is typically developed as part of a schedule version. It conveys information about the schedule trip or trip time to the customer in a publication format.
2	Note Text	• A noteText is associated with a unique noteID. • The noteText is a concise message to explain a part of a schedule. It is displayed to the public. As such, the style should conform to agency print and web style policies. • A note text should be terse; it is recommended that the text not exceed 256 characters. • The note text is added for public dissemination only; it should not be used to replace assignment of day type, trip type, time event type, time type or connection type of any its associated entities.
3	Associations	• There is a many to many relationship between trips and notes, and trip times and notes. As such there should is an association table ascribing the occurrence of a note to a trip or trip time.
4	Note Library	• The note library is a collection of notes that may be used by Trip and Trip Time descriptions. • Each entry will allow an unlimited number of characters.

Conceptual Data Model Description for Note

The requirements for Note are modeled in the CDRM shown in Figure 5-5. The Note data concept includes the Note entity, and its associated entity, Note Association.

SDP XML Schema Model Description for Note

The SDP XML Schema for Note was implemented from the conceptual model without any issues. As a result, there was no need to modify requirements or create special requirements. The SDP XML Schema excerpt of the Note data concept is illustrated in Figure 5-5.

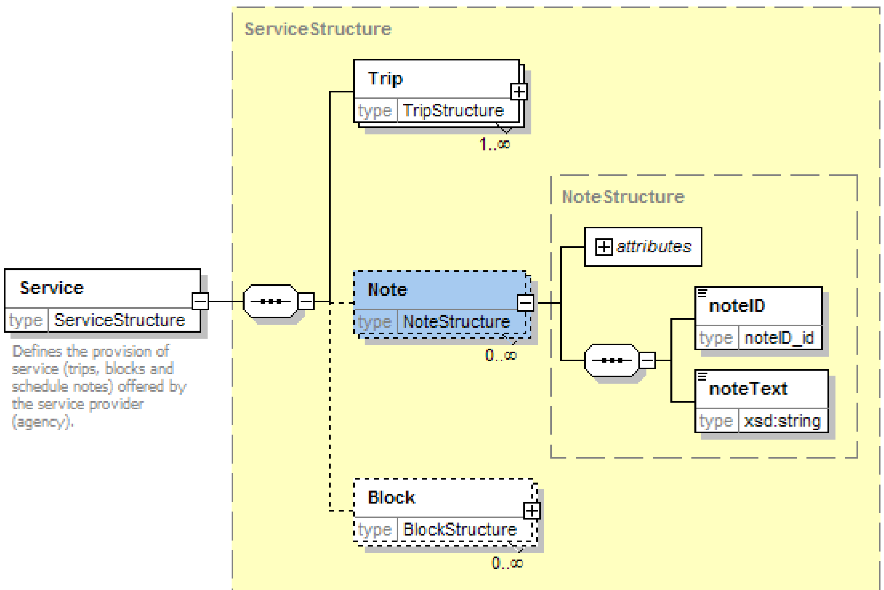


Figure 5-5: SDP XML Schema Excerpt for Note

Data Formats and Guidance for Note

This section describes the format and guidance for each element in the Note data concept described above. The guidance is consolidated into Table 5.2-2, which includes a list of baseline requirements (M for mandatory and O for optional), the element name, the data type and guidance related to element. The guidance provides clarity to the data definition. The first column of each table designates the baseline requirements based on the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet its data needs. The element name corresponds to the related conceptual data model entities and attributes. The type may refer to a native XML type, or declared type in the XML schema. The Guidance column is called “Questions to Ask.” These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

Table 5.2-2: Note Guidance

Required	Element Name	Type	Questions to Ask
M	noteID	noteID_id UNIQUE	A unique identifier to designate the note.
M	noteText	String	The text of the note. This may be used by an application as content for public dissemination.
	effectiveDate	Date	[attribute] The first date on which the note record is valid.
	endDate	Date	[attribute] The last date on which the note record is valid. If the record does not have a specific end date, then use default end date [9999-12-31].

Example of Note Implementation

Notes can be a key source of important information on scheduled trips and trip times. In the example below, Table 5.2-3 lists some of Metro-North Railroad (MNR) Notes (extracted from their timetables).

Table 5.2-3: Example of Notes from Metro-North Railroad

noteID	noteText	Relationship to other SDP elements
A	Amtrak connections	
C	Connecting service	
D	Discharge passengers	maps to TripTime:tripEventType (AlightingOnly)
H	Discharge may depart 5 minutes early	
R	Receive only	maps to TripTime:tripEventType (BoardOnly)
G	Guaranteed ride home	
Bicycle	Bicycle train [includes additional text citing rules related to boarding with a bike…]	
Bus	Bus service connection	

Two of the notes—C and H—are further selected as examples and used to populate the SDP XML Note element, in the fragment of a SDP XML Document shown below.
The two notes excerpted from the table may be rendered to XML as follows:

```
<Service>
<!--Trip elements go here -->

        <Note>
                <noteID> C </noteID>
                <noteText> Connecting service </noteText>
        </Note>
        <Note>
                <noteID>H</noteID>
                <noteText> Discharge may depart 5 minutes early </noteText>
        </Note>
```

</Service>

When Should an Agency Use Notes as Opposed to Other Flags and Codes?

Notes are always acceptable. They are often used to explain special instructions to the public. The drawback of using notes is that they must be manually inserted and may not be well maintained over a long period of time. In contrast, codes and flags are frequently necessary to aid logical processing and implement automated algorithms. They are easier to maintain, and they provide information to downstream applications. So whenever possible, use codes and flags. Definitely add notes when the codes do not convey all the information to the public as well to highlight the information for the downstream user.

Validation Check

Retaining the relationship between the codes and notes is essential. For example, in the third column of Table 5.2-2, several notes relate directly to the code values, e.g., tripEventType (alight and board only). A validation check may be inserted in a native to SDP translator to ensure the data are consistent.

Chapter 6: Transit Network and Related Data Concepts

In This Chapter

- Understand the requirements relating to the Transit Network Branch Model.
- Discover how transit network and related data concepts are used.
- Learn how to apply the Pattern, Transit Point Event and Transit Path Event elements.
- Learn how to convert route segments to Transit Point Events.

Transit Network
The transit network branch describes the physical path over which transit service is delivered. The Transit Network data concepts are designed to describe fixed route operations. It includes the following elements:

- Pattern
- Transit Path

Pattern Definition
A unique, non-branching, ordered sequence of transit paths, timepoints or transit stops to be followed by a transit vehicle in scheduled service for a route in a given direction.

Generic Network Model

Several networks are needed to describe the provision of transit service. Theoretically, in a generic model, the transit network is composed of three "logical" layers, which might be described as the:

- Geographic Network (geography) layer
- Transit Network (transit features and paths) layer
- Transit Service (temporal) layer

These "layers" are shown in Figure 6-1. Each layer builds upon the physical reality or physical locations, such as streets or rail alignments. In the generic model, this base layer is called the *Geographic Network* and it is represented by the geographic database comprised of the digital base map, surveyed points, and other location features.

This chapter focuses on the data concepts related to the Transit Network. The *Transit Network* represents the paths traversed by transit vehicles and is composed of route segments (also known at Transit Paths) and Patterns. The Transit Network layer also includes events (e.g., timepoints, stops) that occur along the path, which may be used to define the Transit Paths and Patterns. The *Transit Service* layer is composed of trips and includes the times when events are scheduled to occur at a particular location (e.g., trip times).

In developing a conceptual data reference model, it is useful to logically separate these layers and store their representations separately. Further, it is critical for transit agencies and their ITS applications, to be able to accurately and unambiguously connect or link data elements between the network layers. For example, bus stops need to be linked to their physical locations on the Geographic Network, to the Patterns in the Transit Network, and to the trip times in the Transit Service layer.

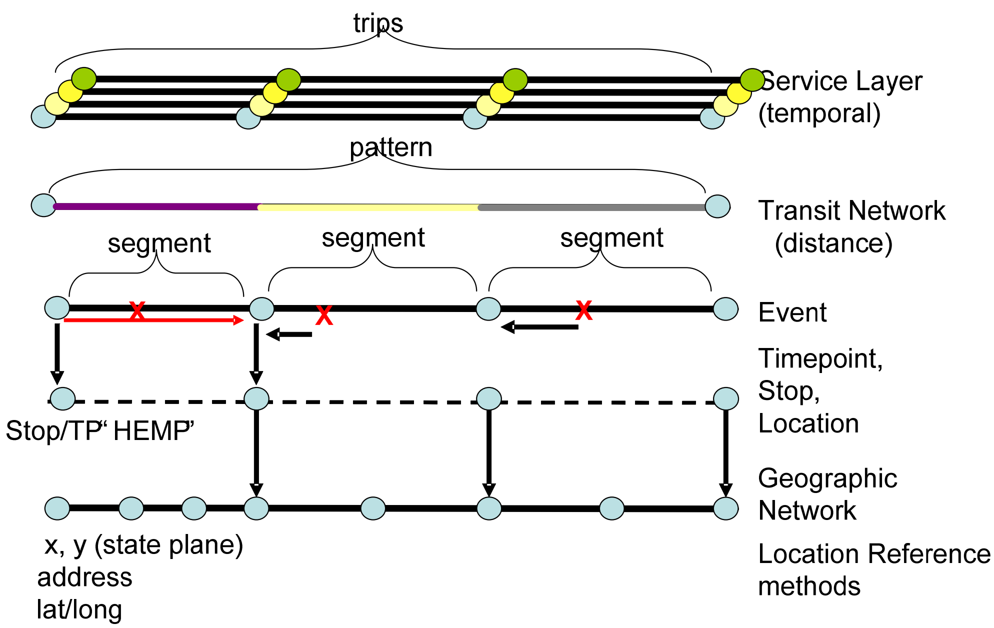


Figure 6-1: Network Layers

The Transit Network, and issues related to implementing the Transit Network data concepts, will be described in this chapter.

Conceptual Data Reference Model (CDRM) Description for Transit Network

A core concept in the CDRM for the Transit Network is the Pattern. Although there are many ways to describe a Pattern, the SDP defines it as a non-branching path that flows in a given direction. That is, a Pattern is directed from origin to destination along a defined set of physical locations. The Pattern may be composed of points or modal paths. Some transit agencies define a Pattern by a series of points, such as bus stops or timepoints. Many other transit agencies use an ordered set of timepoint intervals (TPIs) or route segments to describe a Pattern. If a Route branches, there are common segments that occur over and over again in the set of Patterns that make up the route. Often, transit agencies aggregate and analyze service performance by route segment, for example running time, passenger load, etc.

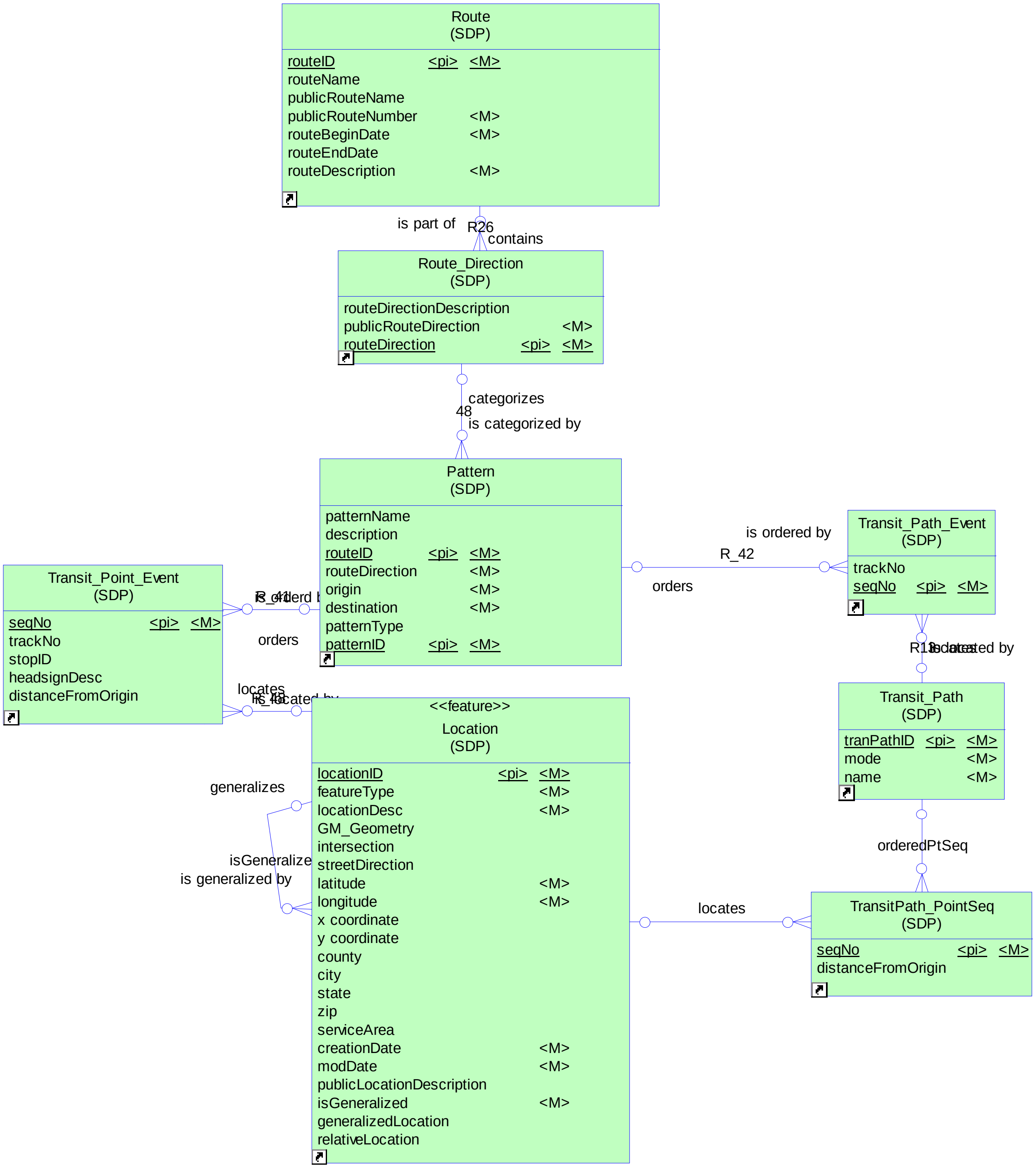


Figure 6-2: Transit Network CDRM Data Concept

The CDRM data model illustrated in Figure 6-2 may be described as follows:

- A Pattern consists of zero-to-many ordered Transit_Point_Events (Locations) or one-to-many ordered Transit_Path_Events. A Pattern is associated with a Route in a given direction (Route_Direction).

- A Pattern may have one or more Trips associated with it. The Pattern is a physical template on which to relate times at specific events. In that way, a Trip may use events (either point or path) upon which to build service descriptions. As such, scheduled service or Trips should correspond to Pattern descriptions. Each Trip should link to an appropriate Pattern, and Trip events should correspond to Pattern events (i.e., Transit Point Events).
- A Transit_Point_Event describes a Location of an activity that occurs along the Pattern. The Point_Event_Type describes the category of events that may occur at the point.
- A Transit_Path_Event (like an instance of a route segment) may be composed of one or more Transit_Paths.
- The Transit_Path is an ordered sequence of Locations (using TransitPath_PointSeq to order each Location along the path). The Transit_Path may be used to represent a route segment or timepoint interval. Although this feature will not be used in this version of the SDP, the entity may be used to define running times or as building blocks for entities that will be used by downstream applications.
- A rail system may have created its own network or rail layer. The rail layer may be represented by the Transit Network layer through the Transit Path element.

As shown in the model above, the Transit Network layer must be associated with location information that is in the geographic network description or digital map, which is used by transit stakeholders and downstream applications. The real world location information may be represented by GPS values, digital base map coordinates, linear measures, or relative locations. In order for the location data throughout the SDP to be consistent, all the geographic information is stored in one place in the SDP model (both XML Schema and CDRM). The coordinate and linear referencing methods are exclusively stored in the Transit Gazetteer elements, specifically, the Location element.

A downstream system will need to know if a Pattern is composed of points or paths, and for that reason, although path definitions are allowed, Transit Point Events *are preferred in the early development and adoption of this specification.* Guidance on implementing a Transit Path and Transit Path Events are also included in order to be comprehensive.

Since the Transit Network Branch must link with other branches or layers of the SDP model, there are several rules included in the Requirements Table (see Table 6-1) to ensure the unambiguous association among the objects that compose each branch.

High Level SDP XML Schema Description for the Transit Network Branch

The rules applied to implement the CDRM as an XML Schema are similar to the other branches. The entity relationships are flattened so that Pattern and Transit Path are hierarchically on the same level. Figure 6-3 depicts the highest level of the Transit Network branch from the SDP XML Schema.

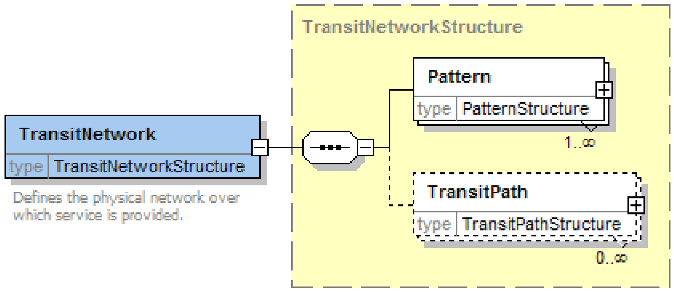


Figure 6-3: Transit Network Model Implemented in the SDP XML Schema

Detailed descriptions of embedded elements in the Pattern and Transit Path elements are discussed in Sections 6.1 and 6.2, respectively.

Section 6.1: Pattern and Related Data Concepts

In This Section

- Learn about the Pattern and Transit Event Data Concepts.
- Learn how to apply the elements in the Pattern and Transit Event Data Concepts.

Pattern Definition
A unique, non-branching, ordered sequence of transit paths, timepoints or transit stops to be followed by a transit vehicle in scheduled service for a route in a given direction.

Transit Point Event Definition
A place where transit service is delivered along the transit network.

Purpose of Pattern Data Concept

A Pattern is used to describe the physical path traversed by a transit vehicle in revenue service. It can either be described by an ordered sequence of points along the path where planned events occur, or as an ordered sequence of segments (Transit Paths). In the SDP, the sequenced list of points or paths are referred to as events. For example, if the event list is describing a sequence of points, the event list may represent an ordered sequence of locations that describe the Pattern, or it may describe an ordered sequence of stops and timepoints. Alternately, the event list refers to a sequenced list of Transit Paths.

The Pattern is the physical alignment where transit service is delivered, while the Trip is the temporal representation of revenue service.

Requirements for Pattern Data Concept

The requirements that drive the Pattern data concept are listed in Table 6.1-1. Some of the requirements for the Transit Point Event and Transit Path Event elements are also discussed in the table.

Table 6.1-1: Requirements for Pattern and Related Data Concepts

#	Category	Requirements
1	Unique/Identity	• A Pattern is unique to a route in a given direction. • A Pattern is referenced by a unique identifier.
2	Geo-spatial Composition as a set of *points*	• A Pattern is a one-way, directed path associated with way points that are used to schedule service. • At a minimum, a Pattern is composed of two (first and last) or more points, and any additional points needed to make it unique. Circular (clockwise and counter clockwise), lollipop and clover leaf configurations may create ambiguity in the path and direction since a Pattern may pass through the same point more than once. To that end, a Pattern should be composed of a sufficient number of ordered geo- or linearly referenced Transit Point Events (may include timepoints and transit stops) to the transportation network to ensure an unambiguous traversal description. • The "way points" consist of a unique, ordered sequence of Transit Point Events such as timepoints, transit stops and events, and additional geo-located points that may be necessary to sufficiently describe the physical path. • A Pattern has an origin and destination. The first and last (origin and

Table 6.1-1: Requirements for Pattern and Related Data Concepts

#	Category	Requirements
		destination) points constitute the termini of the Pattern. Since a trip is a temporal instance of the Pattern, the trip locationBegin and locationEnd should correspond to the Pattern origin and destination, respectively.
3	Geo-spatial Composition as a set of transit *paths* (alternative to set of points)	• Alternatively, a Pattern may be described by a sequence of one or more ordered topologically connected transit paths or street links. • A transit path, sometimes called a *route segment* or *timepoint interval*, is composed of transit point events. (Note: a Transit Path in the SDP expands the definition and use of a Transit Path to mean any carriageway or alignment that is used to deliver transit services.) • A *route segment* is a generic term for a segment that contains an ordered sequence of transit stops, timepoints or events. The directed segment contains at least two points, an origin and destination, and contains sufficient number of points to unambiguously associate the path of travel to the transportation network. • A *timepoint interval* designates a link terminated by two timepoints. Other Transit Point Events such as transit stops may be linearly referenced to the interval. A running time, optionally qualified by time period, day type or other characteristic, may be associated with the interval. (Note: the timepoint interval is not necessarily geo-referenced to a physical path.) • A terminus (destination) of a Transit Path must be coincident with the terminus (origin) of a sequential path in order for the paths to be topologically connected. The paths must flow in the same direction as the previous path. • A Pattern may include an associated "variant." A *variant* is a branch at the beginning, middle or end of a Pattern. ***Variants are not supported in the SDP.*** They should be incorporated into a new Pattern from origin to destination. • A Pattern may be associated with trips that perform "short turns" that do not include an exception or special consideration. • The SDP requires that the points describing a transit pattern (or transit path event) be associated as an event to its respective pattern.
4	Subtypes	• A Pattern may be designated as revenue or non-revenue. • A non-revenue generating Pattern may be a deadhead, pull out, pull in, or non-scheduled path used for training, maintenance or other vehicle journey. Only those non-revenue Patterns that are needed to ensure coordination need be included in the SDP. • Rail transit does not always designate stopping patterns, although the patterns may be derived from a set of trains (or trips) trip times.
5	Transit Point Events	• The Pattern may be composed of an ordered set of Transit Point Events. These events (points) may correspond to: - Transit stops serviced by the route; - Timing points that schedule service; - Operational events that are scheduled to occur during scheduled service. These events may include pull out and pull in points (from depot or layover points that may be neither a transit stop nor a timepoint), fare set change point, headsign change point, short turn,

Table 6.1-1: Requirements for Pattern and Related Data Concepts

#	Category	Requirements
		and other events that may be designated by each local agency. - Physical points to provide shape and reduce path ambiguity.
6	Route Direction	• A Pattern is associated with one of the two Route Directions designated for the route.
7	Attributes	• A Pattern may be associated with a particular service such as regular, express, skip stop, charter, school, etc. • A Pattern may be associated with a particular mode of operation. (This attribute may be better associated with each Transit Path since some construction or other event may necessitate use of an alternate mode for part of the Pattern.) • Schedulers or other staff may sometimes attach one or more notes to a Pattern or event to convey information that applies to all scheduled service on that Pattern, e.g., "construction will occur at this point…"

SDP XML Schema for Pattern, Transit Point Event and Transit Path Event

In implementing the Pattern and related entities into the SDP XML Schema (see Figure 6-4 for XML Schema fragment) a number of rules were applied.

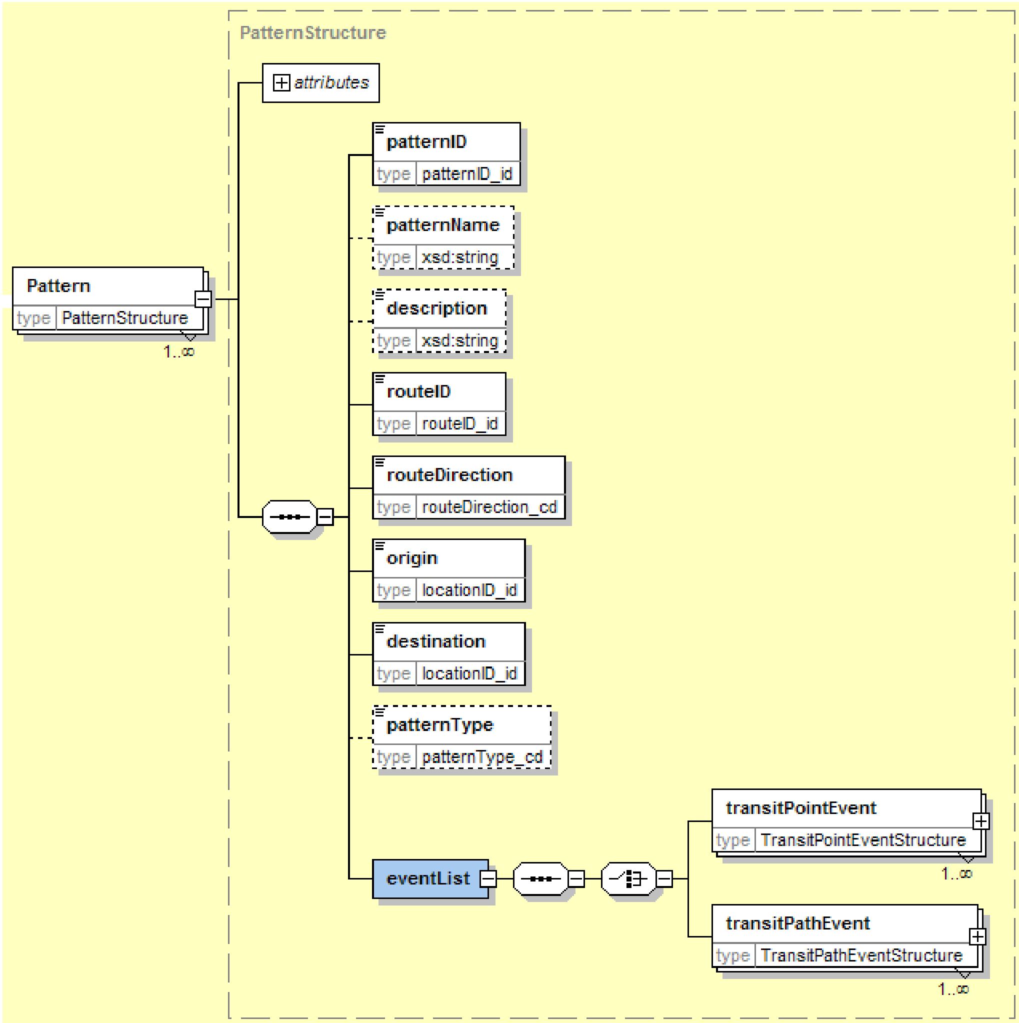


Figure 6-4: SDP XML Schema Fragment of Pattern Element

The rules applied to the Pattern element (in addition to the requirements listed in the Functional Requirements table, Table 6.1-1) include:

- Six mandatory elements are necessary, as shown in Figure 6-4. They include patternID, routeID, routeDirection, origin, destination and an eventList, which is embedded as a child element of Pattern and includes a set of events (either transitPointEvent or transitPathEvent).
- In addition patternID, routeID and routeDirection are required to unambiguously identify each Pattern.
- The routeDirection [first or second] describes a routeDirection value whose native value may be looked up in the Route element routeDirectionList (see Section 4.3).
- The direction of the Pattern's path is derived from origin to destination direction; to ensure the correspondence between the Pattern and eventList elements, the origin (locationID) should be the first event in the eventList and destination (locationID) the last event in the eventList.
- patternType is described as an enumerated value with two values.

- eventList is *either* a set of Transit Point Events or Transit Path Events. This is a CHOICE option in the XML Schema. (Point and path events are discussed later in this section.)
- Attributes, consisting of effectiveDate and endDate, may be used to record the placement dates of the information content.

Transit Path Event / Transit Point Event

Only one structure, TransitPointEventStructure or TransitPathEventStructure, is permitted in the eventList element. At least one or more events should be ordered between the eventList tags. Figure 6-5 shows transitPathEvent and transitPointEvent relative to eventList in the relevant fragment of the SDP XML Schema's hierarchical structure. (See eventList nested in Pattern in Figure 6-5, which is nested in TransitNetwork as shown in Figure 6-4.) The transitPathEvent is linked to the TransitPath, which describes where the path is, via the tranPathID, as shown in Figure 6-4.

The rules applied to the transitPointEvent element include:

- locationID references an identifier of a point that exists in the Location element (Transit Gazetteer branch).
- trackNo and stopID are optional feature references. They may be found using a join through the locationID (with the Track or TransitStop elements). The data is included in the transitPointEvent because many downstream applications use this information directly from the transitPointEvent description.
- The stopID may also refer to a platformNo.
- Similar to distance in TransitPath, distanceFromOrigin is the linear, as-traveled distance from the origin of the Pattern to this event location. The element includes an attribute of unit wherein the unit of measure—meters or feet—may be designated. The default value when no attribute is present is "feet."
- seqNo is the order of the event in the list.
- headsignDesc is the content of the headsign at the event. By associating the headsign with the event, and not the pattern, a series of messages, such as "via a destination," or next trip information, may be changed during a journey or prior to the end of a path. *The first event should contain the default headsign message.*

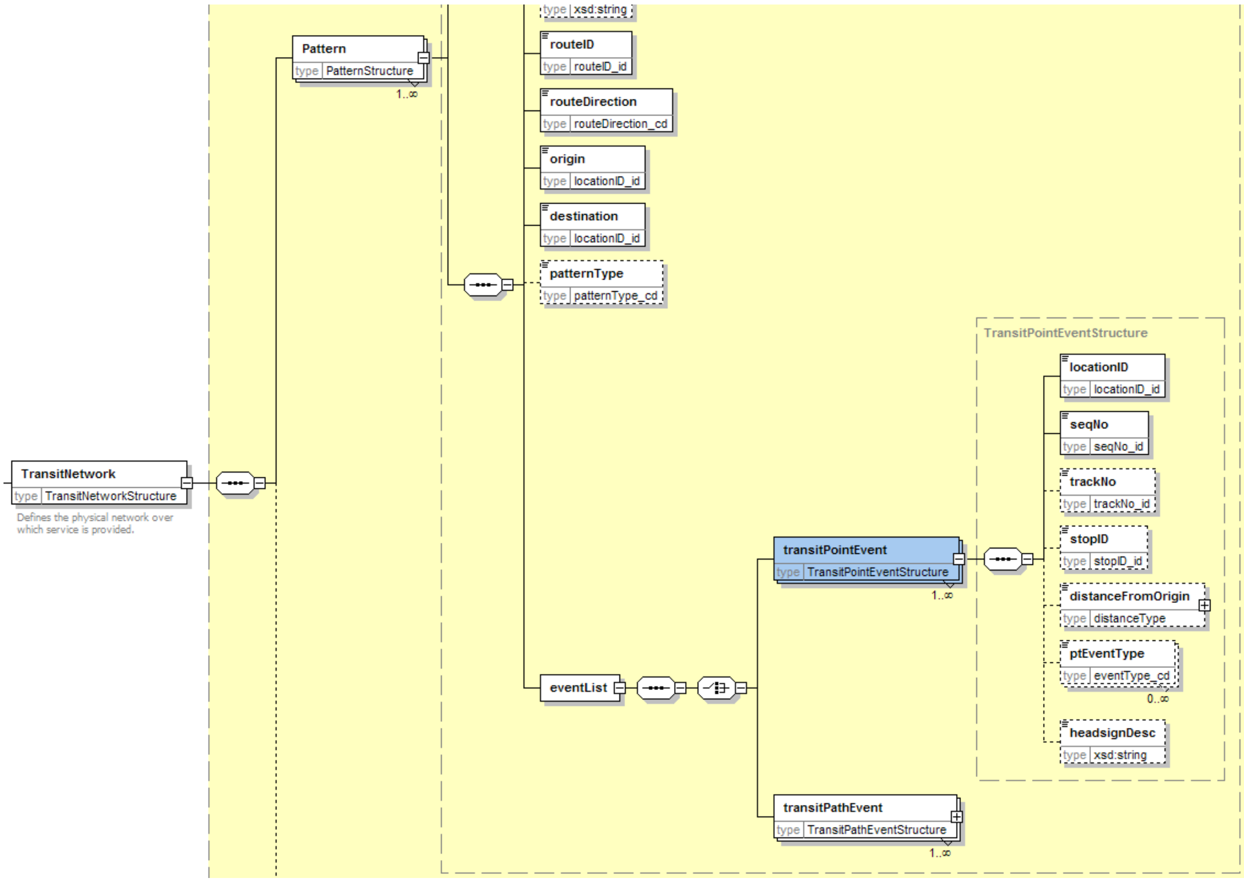


Figure 6-5: SDP XML Schema fragment showing eventList CHOICE elements transitPointEvent and transitPathEvent

The transitPathEvent element conforms to the requirements listed in Table 6-1. The rules applied to the transitPathEvent element include:

- At least one or more transitPathEvents should be ordered between the eventList tags.
- The linkage between the transitPathEvent and the actual Transit Path (see Section 6.2) is through the tranPathID.
- seqNo is the order of the transitPathEvent in the eventList. It determines the sequence of the Transit Paths in the Pattern.
- trackNo is an optional feature reference. It may be found by using a join through the trackNo with the Track element described in the Transit Facilities Branch (See Figure 8-8 in Chapter 8).

Data Formats and Guidance for Pattern, Transit Point Event and Transit Path Event

This section provides descriptions and guidance for the elements associated with Pattern in Table 6.1-2, for Transit Point Event elements in Table 6.1-3, and for Transit Path Events in Table 6.1-4. The guidance is consolidated into tables that include a list of baseline requirements (M for mandatory and O for optional), the element name, the data type and guidance related to element. The guidance provides clarity to the data definition. The first column of each table designates the baseline requirements based on the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet its data needs. The element name corresponds to the related CDRM entities and attributes. The type may refer to a native XML type, or declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

Table 6.1-2: Pattern Structure and Guidance

	Element Name	Type	Questions to Ask
	Pattern		
M	patternID	patternID_id UNIQUE	What field in the native data is used to designate a Pattern? Are Pattern identifiers unique throughout a route? If no, then add new Pattern identification values to ensure that each Pattern in a route is uniquely identified.
O	patternName	string	Do Patterns have names?
O	description	string	Description of the Pattern. The field may not be needed for most downstream applications.
M	route	routeID_id	The route identifier associated with the Pattern. This field is validated for its existence.
M	routeDirection	routeDirection_cd	The direction of travel. Use standard code 'first' or 'second' and designate publicRouteDirection in the Route.routeDirectionList element.
M	origin	locationID_id	The originating location of the Pattern. This field should match the first entry in the transitPointEvent.locationID element. The locationID should exist in the Location element.
M	destination	locationID_id	The location of the destination of the Pattern. This field should match the last entry record in the eventList (that is transitPointEvent.locationID). The locationID should exist in the Location element.
O	patternType	patternType_cd	A classification for a Pattern. Acceptable values include revenue and non-revenue. Current downstream applications do not need non-revenue patterns or trips.
M	eventList	Choice of ordered list of TransitPointEvent or TransitPathEvent	The elements in this entry are the events that occur along the Pattern. A commuter rail system where each train (trip) may stop at a unique set of stations/events, only one Pattern for each origin-destination pair (along non-branching paths) need be described.

Table 6.1-3: Transit Point Event Elements and Guidance

	Element Name	Type	Questions to Ask
	Transit Point Event		
M	location	locationID_id	A reference to the location where the event occurs. The point may be any type of transit feature, physical point, timepoint, transit stop or facility, or other event (e.g., change headsign, trigger location). The locationID should exist in the Location element.
O	trackNo	trackNo_id	For trains, the trackNo on which the event occurs.
O	ptEventType	ptEventType_cd [0..∞]	The type of event that occurs. More than one event may occur at a single point, for example, exiting a bus stop zone and checking out of a Transit Signal Priority zone may be co-located.
O	stopID	stopID_id	If appropriate, the stop identifier or platform number associated with the event.
O	distanceFromOrigin	float	The distance from the Pattern origin. Although this information is not universally collected, it is needed to reduce ambiguity by many downstream applications, and thus is a key field for the SDP. Also includes optional attribute unit= [feet or meters]. Default when no attribute is present is feet.
M	seqNo	integer	The sequence number of the event along the Pattern. This number may be derived if the distanceFromOrigin is defined.
O	headsignDesc	string	Although this field may be included multiple times, the field may also contain additional fields to be used throughout the Pattern, for example, headsign information such as "via Brooklyn/Flatbush" or "Go Mets."

Table 6.1-4: Transit Path Event Elements and Guidance

	Element Name	Type	Questions to Ask
	Transit Path Event		
M	tranPathID	tranPathID_id	A reference to the link or path identifier. This identifier is associated with the Transit Path definition of the geographic path or traversal. The identifier should exist in the Transit Path element.
	trackNo	trackNo	If this is a rail alignment, then the track identifier may be included here.
M	seqNo	integer	An index that orders the transitPathEvents and by direct relationship, the segments or Transit Paths, in a Pattern. The Transit Pattern has direction, so by default, this segment has direction. The ordered set of segments should be topologically valid, that is, the first point of this segment should be the same as the last point of the previous segment.

Usage and Example of Transit Pattern

Example 1: A Pattern with fields completed using Transit Point Events.

This example shows an excerpt from an SDP XML Document, which describes a pattern from Route 3210 (N1 Jamaica-Elmont-Hewlett from Long Island Bus). It was derived from an ordered set of route segments to define a single set of point events (transitPointEvent) from origin to destination of the Pattern.

This Pattern listing includes the mandatory elements patternID, routeID, routeDirection, origin, destination and eventList. The eventList tags enclose an ordered list of transitPointEvent elements. In turn, the transitPointEvent tags distinguish the elements in sequential order from seqNo 1 to the last seqNo 53. Notice that the origin and destination elements match the first and last transitPointEvent locationID elements (respectively). The distanceFromOrigin is a floating point number. As a general rule in the Long Island Bus data set, the native data does not insert an absolute 0.0 value, and so the first point will appear as 1E-005. Values for origin, destination and locationID come from the first and last transitPointEvent elements.

```
<Pattern>
 <patternID>102</patternID>
 <routeID>3210</routeID>
 <routeDirection>second</routeDirection>
 <origin>4918</origin>
 <destination>225</destination>
 <eventList>
   <transitPointEvent>
     <locationID>4918</locationID>
     <seqNo>1</seqNo>
     <distanceFromOrigin units=meters>1E-005</distanceFromOrigin>
     <headsignDesc>N01 HEWLETT DIRECTION South</headsignDesc>
   </transitPointEvent>
   <transitPointEvent>
     <locationID>553</locationID>
     <seqNo>2</seqNo>
     <distanceFromOrigin units=meters >549.64</distanceFromOrigin>
     <headsignDesc>N01 HEWLETT DIRECTION South</headsignDesc>
   </transitPointEvent>
…
   <transitPointEvent>
     <locationID>581</locationID>
     <seqNo>51</seqNo>
     <distanceFromOrigin units=meters >13863.22</distanceFromOrigin>
     <headsignDesc>N01 HEWLETT DIRECTION South</headsignDesc>
   </transitPointEvent>
   <transitPointEvent>
     <locationID>614</locationID>
```

```
      <seqNo>52</seqNo>
      <distanceFromOrigin units=meters >13999.28</distanceFromOrigin>
      <headsignDesc>N01 HEWLETT DIRECTION South</headsignDesc>
    </transitPointEvent>
    <transitPointEvent>
      <locationID>225</locationID>
      <seqNo>53</seqNo>
      <distanceFromOrigin units=meters >14469.5</distanceFromOrigin>
      <headsignDesc>N01 HEWLETT DIRECTION South</headsignDesc>
    </transitPointEvent>
  </eventList>
</Pattern>
```


In the example above, the Route element defines routeDirection “second” as the South, as shown in Figure 6-6.

Figure 6-6: Route Direction Description


```
<routeDirectionList>
  <routeDirection>second</routeDirection>
  <routeDirectionDescription>bound=1 direction=4 tcip=2</routeDirectionDescription>
  <publicRouteDirection>South</publicRouteDirection>
</routeDirectionList>
```

Section 6.2: Transit Path and Related Data Concepts

In This Section

- Learn about the Transit Path Data Concept.
- Understand how to apply the elements in the Transit Path and relate it to the Transit Pattern's Transit Path Event.

Transit Path Definition
A transit path is a linear section and geographic representation of the Transit Network which is designed for the movement of Public Transit Vehicles.

Purpose of Transit Path Data Concept

A Transit Path is a description of a geographic alignment or guideway over which transit service is delivered. It may be the description of a route segment, timepoint interval, rail track, walking path, bus only lane or other geographic feature. One of its primary purposes is to use the Transit Path as a route segment that is used in a Pattern as an element in the eventList, that is, one of a sequence of transitPathEvents. Although the Transit Path element is optional in general, it becomes mandatory when the Pattern is described by transitPathEvents. In that case, the Transit Network is composed of Transit Paths either along a street network, private road or a transit-only path, such as a rail tracks, bus-only lanes or guideways. It may also be used to describe a "patch" to a transportation network over which a transit vehicle traverses. The Transit Path, by definition, is directed and is composed of termini (end points), and optionally, an ordered set of points along the path between the termini.

The requirements pertaining to the inclusion of a Transit Path in a Pattern are delineated in Table 6.2-1.

SDP XML Schema for Transit Path

Figure 6-7 shows the SDP XML Schema fragment for the Transit Path. In implementing the Transit Path and related entities into the SDP XML Schema, a number of rules were used to implement the data concept. These rules and assumptions include:

- A unique identifier is used to index the Transit Path. A name may optionally be associated with the Transit Path.
- All Transit Paths are directed, starting at the origin and ending at the destination. The origin and destination refer to a Location identifier (locationID).
- An ordered set of points (between the origin and destination) should be included that describes the physical path of the Transit Path.
- The value of the mode element is a code that defines the service mode (e.g., type of transit vehicle) that uses the Transit Path.
- The distance is the linear distance (floating point) from origin to destination. Distance includes an attribute that designates unit of measure. Unit values include feet and meters. Default value (when not present) is "feet."
- Attributes, comprised of effectiveDate and endDate, may be used to record the placement dates of the information content.

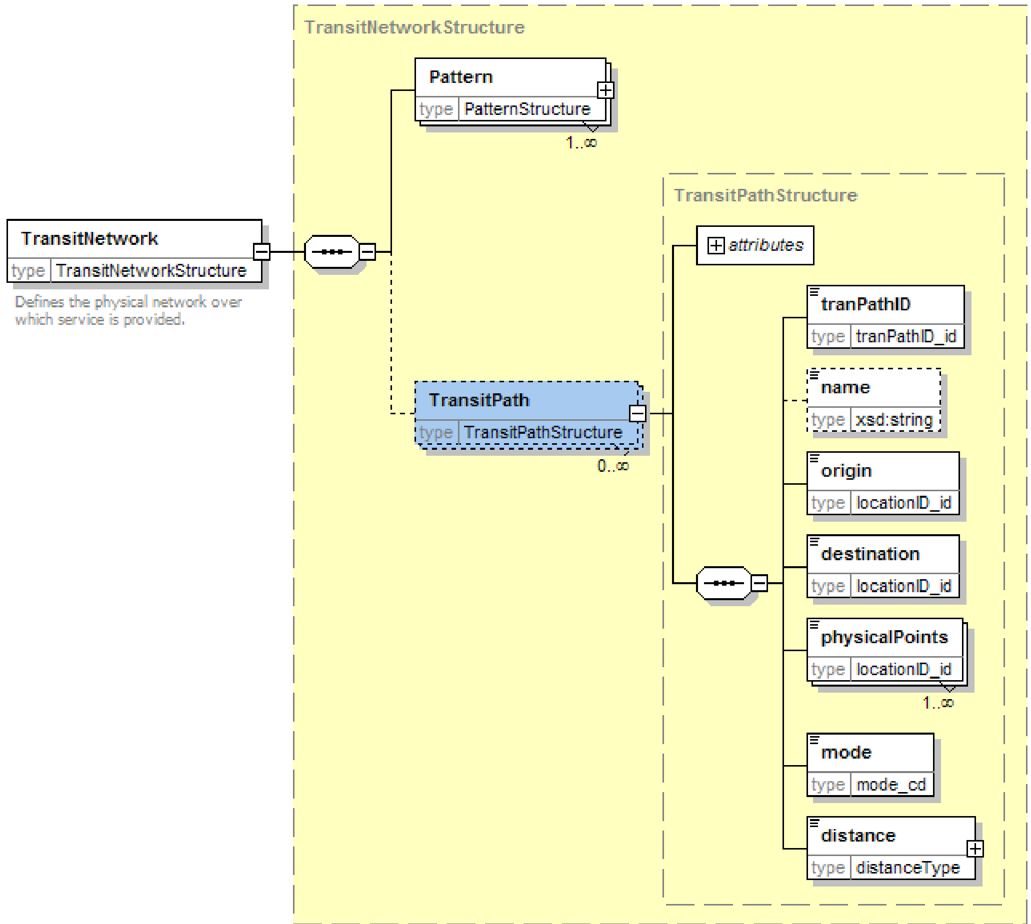


Figure 6-7: SDP XML Schema Fragment of TransitPath

Detailed Data Formats and Guidance for Transit Path

This section explains the formats and guidance associated with each element in the Transit Path data concept illustrated in Figure 6-7 above. The guidance is consolidated into Table 6.2-1, which includes a list of base line requirements (M for mandatory and O for optional), the element name, the data type and guidance related to element. The guidance provides clarity to the data definition. The first column of each table designates the baseline requirements based on the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet its data needs. The type may refer to a native XML type, or declared type in the XML schema. The Guidance column is called “Questions to Ask.” These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document.

Table 6.2-1: Transit Path Elements and Guidance

	Element Name	Type	Questions to Ask
	Transit Path		
M	tranPathID	tranPathID_id UNIQUE	A unique identifier for the Transit Path.
O	name	string	If the Transit Path has a name, insert it here.
M	origin	locationID_id	The starting point of the segment (Transit Path). The locationID should exist in the Location element.
M	destination	locationID_id	The ending point of the segment (Transit Path). The locationID should exist in the Location element.
M	physicalPoint	ordered set of locationID_id	The ordered or directed set of points (locationID) starting from the origin and ending with the destination. Sufficient number of points should be included to ensure that the physical path is unambiguously defined. If the path loops or backtracks, additional points may be needed to clearly define the physical path of travel. The locationIDs used should exist in the Location element.
M	mode	mode_cd	The type of vehicle or service that uses the Transit Path. For example, the path may be a ferry lane for ferry service, caternary lines for streetcar service, or tracks for commuter rail or subway service.
M	distance:	float	The linear distance of the Transit Path. Also include optional attribute unit [feet or meters]. Default when no attribute is present is feet.
O	effectiveDate	date	[attribute] The date the record was placed or inserted.
O	endDate	date	[attribute] The date the record expires or becomes obsolete. Default value is 9999-12-31.

Usage and Examples of Transit Path Element

A route segment may be represented as a Transit Path. The SDP XML Document fragment below describes Long Island Bus Pattern 116 and its related route segments: 100-103 (first to third segment), and fifth and last segment of Pattern 116, Route 3210.

- The route segment (tranPathID 100) has 5 defined points that ranges over a linear distance of 1526.00 meters.
- The route segment (transPathID 101) has 7 defined points that ranges over the linear distance of 4912.89 meters.
- The route segment (tranPathID 102) has 8 defined points that ranges over a linear distance of 1,736.67 meters.
- The route segment (tranPathID 105) has 3 defined points that ranges over a linear distance of 749.21 meters.

These route segments are ordered as a transitPathEvent in Pattern 116 in its eventList. The example displays the order of elements based on the SDP XML Document. In the SDP, the Pattern elements are listed prior to TransitPath.

```
<Pattern>
  <patternID>116</patternID>
```

```
 <routeID>3210</routeID>
 <routeDirection>second</routeDirection>
 <origin>232</origin>
 <destination>228</destination>
 <eventList>
   <transitPathEvent>
     <tranPathID>100</ tranPathID >
     <seqNo>1</seqNo>
   </transitPathEvent>
   <transitPathEvent>
     < tranPathID >101</ tranPathID >
     <seqNo>2</seqNo>
   </transitPathEvent>
…
   <transitPathEvent>
     < tranPathID >105</ tranPathID >
     <seqNo>5</seqNo>
 </eventList>
</Pattern>
…
<TransitPath>
        <tranPathID>100</tranPathID>
        <name>Route 3210; patternID 116; south; segment 1 </name>
        <origin>232</origin>
        <destination>234</destination>
        <physicalPoints>5249</physicalPoints>
        <physicalPoints>556</physicalPoints>
        <physicalPoints>5250</physicalPoints>
        <mode>MB</mode>
        <distance units=meters>1526.00</distance>
</TransitPath>
<TransitPath>
        <tranPathID>101</tranPathID>
        <name>Route 3210; patternID 116; south; segment 2 </name>
        <origin>540</origin>
        <destination>5257</destination>
        <physicalPoints>566</physicalPoints>
        <physicalPoints>539</physicalPoints>
        <physicalPoints>5254</physicalPoints>
        <physicalPoints>537</physicalPoints>
        <physicalPoints>524</physicalPoints>
        <mode>MB</mode>
        <distance units=meters>4912.89</distance>
</TransitPath>
<TransitPath>
        <tranPathID>102</tranPathID>
```

```
        <name>Route 3210; patternID 116; south; segment 3 </name>
        <origin>235</origin>
        <destination>4918</destination>
        <physicalPoints>661</physicalPoints>
        <physicalPoints>518</physicalPoints>
        <physicalPoints>517</physicalPoints>
        <physicalPoints>516</physicalPoints>
        <physicalPoints>515</physicalPoints>
        <physicalPoints>4316</physicalPoints>
        <mode>MB</mode>
        <distance units=meters>1736.67</distance>
</TransitPath>
…
<TransitPath>
        <tranPathID>105</tranPathID>
        <name>Route 3210; patternID 116; south; segment 5 </name>
        <origin>227</origin>
        <destination>228</destination>
        <physicalPoints>5537</physicalPoints>
        <mode>MB</mode>
        <distance units=meters>749.21</distance>
</TransitPath>
…
```

Chapter 7: Transit Gazetteer and Related Data Concepts

In This Chapter

- Explore the Transit Gazetteer Branch of the SDP Model and its Location Table.
- Discover how transit network and related data concepts are used.
- Learn how to apply Pattern and Transit Point Event.
- Learn how to convert route segments to transit point events.

Transit Gazetteer Definition

A transit gazetteer is a geographical dictionary and reference for information about places and place-names associated with a specified data set such as the SDP document or TSDEA registered schedules, landmarks or other places of interest related to the transit network.

Elements include:

- Location
- Timepoint
- Transfer Cluster

One of the key elements of the SDP's gazetteer is a Location Table, which facilitates access to location data and simplifies maintenance.

Purpose of a Transit Gazetteer Model

Location related information is critical component of an effective transit schedule, hence its inclusion in the SDP. Generally, a schedule describes when transit service will be provided at a specific location. Time and place are important information for schedule users. The schedule includes locations where time is measured, such as where transit vehicles embark or debark, where to transfer from one service/mode to another, as well as, the locations of transit facilities.

The gazetteer concept is used by many industries to describe the location of places that are used by their services. The SDP uses this concept as a geographic dictionary and reference for information about transit places and place-names. Using a transit gazetteer with a Location Table mitigates several location referencing management issues facing most transit agencies. Among its benefits, the transit gazetteer:

- Creates a single location to find and use various types of transit features;
- Provides an extensible, single location to associate, update and propagate changes to several types of location reference methods;
- Isolates the geographic reference maintenance from the domain data maintenance schedule;
- Eliminates ambiguity inherent in the location measurement by indexing “location” using a unique, sequential number; and
- Enables an accurately measured position to be linked to a more generalized location.

Specifically, a Gazetteer with a Location Table supports the management of spatial information necessary for defining the transit network. It supports flexible, more efficient, long-term data management, and the ability to define other transit features types “on the fly” that may be needed to further describe the transit network.

Typically, all transit places are included in a Gazetteer. The Location Table, which is a key element in the Gazetteer Branch, serves to link “transit places” such as timepoints, stops and other facilities, to the transportation network. The Transit Gazetteer includes the Timepoint and Transfer Cluster (a place where transfers between transit service is recommended). Since the transit facilities model is particularly complex, it was given its own branch outside the Gazetteer in which to describe places related to facilities (see Chapter 8: Transit Facilities Branch). Discussion on the Transfer Cluster is included in Chapter 10: Advanced Topics and covered in Section 10.4: Transfers.

Issues Affecting the Design of the Gazetteer

Past practices and existing systems at transit agencies have led to a variety of ways of representing location data, and have exposed a variety of problems, which have challenged the implementations and operations of ITS and Geographic Information Systems (GIS). The design of the Gazetteer is intended to avoid or minimize these challenges.

The issues are discussed here to give the reader a better understanding of how certain approaches to creating and organizing location data can cause transit agencies problems, why the SDP is designed as it is, and to provide insight for the data conversion effort.

Issues With Location Referencing

The ability to locate the transit network upon which transit service is scheduled and to use the location information for transit features is an important need for most downstream applications. Four commonly encountered issues with location referencing will be discussed.

Often, location is handled as an attribute of an entity. This approach is not sufficient to handle the existing range of location referencing methods that transit operators need and that downstream applications use. Similarly, this approach will not support existing and future requirements for locating places of transit-based "events" and maintaining the connections among transit and transportation network data.

Many transit data models merge the transit and spatial feature concepts into a single *entity*. So features such as transit stops that may only need address and intersection may not include a reference compatible with a Cartesian coordinate used by a timepoint, and those events along a pattern used by the on-board navigation system may only use the spherical coordinates (e.g., WGS 84). As such, these equivalent locations may be dispersed throughout a data set and are incompatible. The consequence is that location may not be used as an integrating characteristic.

Further, different data sets may require different, yet equivalent, location referencing methods. For example, the same place, New York Penn Station, may be described by a variety of location referencing methods by different transit agencies:

- 25th stop (out of 33) in an express route from Staten Island through NJ (the S22X).
- 50 feet west on the south side from the corner on 34th Street intersecting with 8th Ave
- near side on 34th Street/at 8th Avenue
- about 5% along the 34th Street block between 8th and 9th Avenues
- 310 W 34th St, New York, NY 10001
- Trip123 lat/long -73.993559, 40.752190
- NY Box-id lat/long -73.993680, 40.752193

The logical location may appear as different locations, although they are equivalent.

The SDP approach provides a single place (the Transit Gazetteer) to access, share and store equivalent locations. Downstream applications may select the most appropriate location referencing method to meet its needs. These reference methods are grouped into a single instance so that the equivalence is obvious. Furthermore, information is stored in a single table to manage changes to the data.

Negotiating the Differences Between the Transit and Geospatial Domains

A number of issues may arise for transit when describing the spatial attributes of a transit feature. From a spatial location perspective, transit service may be viewed as operating on a changing and scalable environment. As a result of this complicated spatial operating environment and other data-related factors, the following categories of issues, which could affect the SDP data, will be discussed:

- Location Reference Method Equivalences
- Generalized Versus Accurate Location References
- Heterogeneity of Transit Features and Their Use as a Transit Network Building Block
- Coordination of Data Set Maintenance Schedules

These issues may arise within a single transit agency and can pose even greater challenges and barriers in a multi-agency environment.

Location Reference Method Equivalences

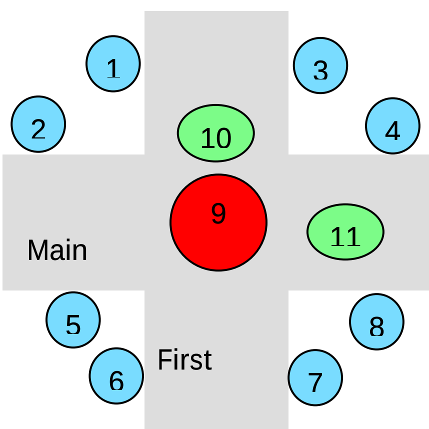
A number of challenges may arise when trying to determine if the location of transit features and events are the same. Transit applications may require the integration of multiple data sources that have very little in common other than proximity. In an operating environment that includes multiple transit agencies, an example would be bus stops that are used by more than one agency, but have different identifiers and different ways of measuring location. Moreover, the location references may be derived from heterogeneous sources, which vary in accuracy, scale, resolution and precision, sometimes significantly.

The *equivalence* of two location reference methods may exist (that is, the location may be the same but expressed in different formats), but the association may not be made even by spatial analysis tools. Sometimes painstaking human analysis is the only means of associating "equivalent location references." These relationships must then be documented in an "association" or "lookup" table. This will be the case for the SDP and Transit Schedule Data Exchange Architecture, where collocated stops will be represented by different agencies using heterogeneous location referencing methods, and each time the source data is updated, the lookup table will require a readjustment.

The human intervention may occur even within an agency that standardizes on a single base map. For example, when a change occurs to a frame of reference such as a base map, all the location references that rely on that frame of reference must also be changed. Often, these attributes are dispersed throughout the agency's data model. Because other location references are dependent on the frame of reference, a significant number of other records will need to be updated.

Generalized Versus Accurate Location References

Along with the varied types of location references and the heterogeneous sources from which the attributes are generated, the levels of accuracy for location references can vary greatly between geographic areas, agencies and applications. Some agencies provide precise and accurate locations while others provide "generalized," low resolution or approximate locations. Figure 7-1 shows a variety of different ways that transit agencies locate bus stops near an intersection. For example, point 9 may represent locations designated as points 1 through 8; point 10 may represent points 1, 3, 6 and 7, which could be stops for a route that provides service along 1st Street; and point 11 may represent points 2, 4, 5 and 8, which could be stops for a route that provides service along Main Street. One agency may collect data by performing a field survey of bus stop locations, measured to an accuracy of 20 centimeters (e.g., points 1 through 8), another may select the location from an orthophoto with 20 meter accuracy (e.g., points 10 and 11), while yet another may select the nearest intersection as the location of the stop (e.g., point 9). For some applications, an intersection reference may be sufficient, while other applications require sub-meter accuracy.



Intersection Designated Points

1. 1st & Main nearside (NS) NW
2. Main & 1st farside (FS) NW
3. 1st & Main FS NE
4. Main & 1st NS NE
5. 1st & Main NS SW
6. 1st & Main FS SW
7. Main & 1st NS SE
8. Main & 1st FS SE
9. Main & 1st (represents 1-8)
10. First & Main (rep. 1,3,6,7)
11. Main & First (rep 2,4,5,8)

Figure 7-1: Intersection with Generalized and Accurate Point References

This problem shares similar challenges to the "equivalence" discussion above. A generalized location, like the centroid of Jamaica Station will need another generalized location to identify the AirTrain, NYCT subway and LIRR terminals, and additional location references to identify the platforms within each station area, and so on. A mechanism is needed to qualify the measurement, not to identify the accuracy, but rather to indicate whether the location approximates or pinpoints the physical object.

The SDP and the Gazetteer provide a solution to these problems. The SDP includes a Boolean element called "isGeneralized." When a place is equivalent to points 9, 10 or 11, then the field is assigned a value of "true." When a place is equivalent to points 1 through 8, then "isGeneralized" is assigned a value of "false."

Generalized and Accurate Location Data Approach

The SDP has developed an approach for dealing with the challenge of identifying and organizing location references that are associated with other locations. For example, multiple stops can be associated to a timepoint or facility, and multiple stops can be part of a larger facility (e.g., Jamaica Station bus stops, platforms and AirTrain platforms). This issue is exacerbated when the data originates from multiple sources, which occurs when there are multi-agency facilities such as those in the downstate NY region.

The recommended approach associates specific or accurate locations with a generalized location. A transit feature should only reference at most one locationID at a time. Incorporating a generalized location may cause several accurate location instances to diverge with respect to the generalized location. As such, the general location should become a managed instance to which the accurate location instances may be associated with, as depicted in Figure 7-2.

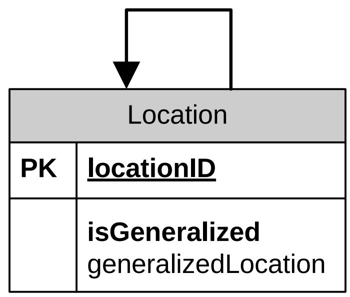


Figure 7-2: Partial Location Table Where an Associated Location May Be Associated With a Generalized Location

Figure 7-3 is an example of how to implement an instance of a self-referencing Location Table, that is how generalized locations and isGeneralized elements may be applied.

For a self-referencing set of locations as shown in Figure 7-3, the following definitions apply:

- locationID is a unique, sequential, unambiguous integer.
- isGeneralized is a Boolean (true/false); when true the value indicates that this field is approximate, centroid or generalized, and that another Location record points to this locationID as a generalized location.
- generalizedLocation is a (foreign key) locationID of a generalized location to which an associated locationID is inserted.

Figure 7-3 uses the diagram in Figure 7-1 as a reference. The scenario describes location 9 as a generalized location with associated locations 1, 2 and 3.

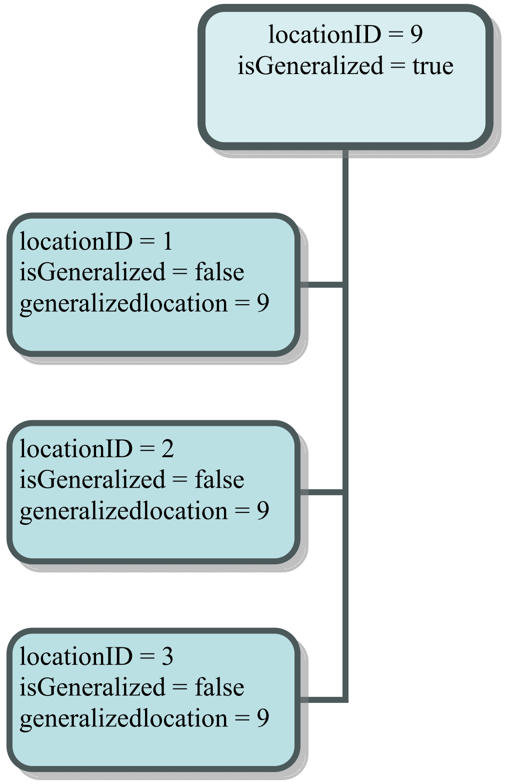


Figure 7-3: Example Instance of a Self Referencing Location Table

Location 9 is assigned a value of "true" for isGeneralized; generalizedLocation is not relevant, so it is not included in the partial list of elements. In the other instances of Location, each box depicts a record for the remaining locations. They are not generalized (rather they are specific), and they reference locationID = 9.

Heterogeneity of Transit Features and Their Use as a Transit Network Building-Block

The transit feature is the building block of a transit network. The industry uses different models for describing the transit network (i.e., patterns and route segments), however, they all encounter the issue of heterogeneous feature types, challenging the ability to integrate these features in a single, ordered list. The pattern event, as described in Section 6.1 may be comprised of one or more transit features such as transit stop, timepoint, physical path location along a route, or other trigger that supports service operations. The variety of types of transit features and the potential need to integrate new event types necessitates a single method for describing and ordering these features into a list of common data types.

Feature Type
Transit Features are objects that represent real world public transport phenomena. [GOS 7d] Transit Stop, Timepoint, Transit Facilities, Track, Transit Path are examples of a feature type.

Relative Location
Relative location is the linear relationship of a transit place such as a Transit Stop to the transportation network.

In the SDP, the Location table serves as a parent type, while all event types inherit location attributes stored in the Location element. Figure 7-4 helps illustrate a problem that transit agencies face when they want to integrate different features in a single ordered list of locations (e.g., they want to know the order of appearance along a path of timepoints, trigger points and bus stops). In the example, a pattern, described as a "unique, non-branching, ordered sequence of transit paths, timepoints, [events] or transit stops to be followed by a transit vehicle in scheduled service," incorporates a single point that may be specialized to reference one of many different types of events (timepoints, bus stops, fare set change locations, headsign change locations, and transit signal priority trigger locations).

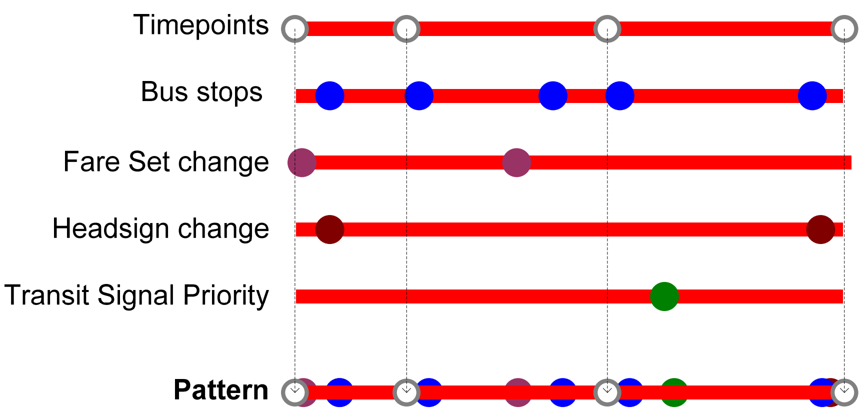


Figure 7-4: Pattern Overlaid with Various Feature Type Locations

Coordination of Data Set Maintenance Schedules

Downstream applications require different location reference methods, which, in turn, require different kinds of data fields in an entity. Some of the data fields come from data sets with different maintenance schedules. The *spatial data* maintenance plan, which may be driven by an external organization, may be different from the *transit domain* data maintenance plan, complicating the maintenance of an **entity** record.

For example, an update to the base map, upon which the location reference is based, may impact the accuracy of the location of a bus stop. The base map update may occur semi-annually. The bus stop inventory which is geo-located to the base map may be updated on a quarterly basis. The specific dates may not correspond, requiring an update of the bus stop inventory at least six times per year (while the schedule version may only be updated quarterly). In addition, all downstream applications that use the bus stop inventory also need to update their base map to correspond to the updated inventory; they do not always correspond to the base map update either. The maintenance and update procedures become complex as the number of data sets and applications that support entities with location references grows.

The design of the SDP minimizes these problems related to maintenance schedules by separating location references from the transit domain entity. As a result, the update of the location reference is modularized; a spatial data maintenance update is of no great impact to the bus stop inventory and the downstream applications that use older or updated information.

Feature Types Supported by the SDP

The primary purpose of the SDP Gazetteer is to clearly define location-related key features that support the construction of the transit network. In the current SDP model, several features are defined by entities, including the following: transit facility, timepoint, transfer cluster, and transit stop. Other supported feature types include: event locations, shape or trace points, as well as points of interest (POS). Landmark or point of interest information is supported, but not necessarily included as a feature. The SDP enables the user to define additional feature types that may be necessary for downstream applications or to support their internal requirements. The model is extensible through the mandatory featureType field. (Note: a specific inventory is not included in the agency SDP document, the landmark inventory should be available from the Web Data Maintenance System–WDMS).

Regional discussions identified other important features that will likely be needed in the future. For example, some ITS applications require different types of event trigger indicators such as for fare-set changes, and interior annunciator/sign triggers. In the discussion on “Heterogeneity of Transit Features and Their Use as a Transit Network Building Block,” the discussion identified “physical path locations” to ensure that the physical path of the pattern is unambiguously defined (particularly when a vehicle loops and backtracks through parking lots, one way streets and highway ramps). The Gazetteer approach supports location information for existing and future transit features through an expanded list of values for the feature type.

Requirements for Location Table Data Concept

The requirements that drive the Location Table data concept are included in Table 7-1. These requirements are derived based on the needs of the downstream applications that use schedule and related data concepts.

Table 7-1: Requirements for Location Table and Related Data

#	Category	Requirements
1	Unique identification, naming conventions, and references	• A place or transit feature is any location within the transit network or service area that may be described by a physical location. • A place or location is indexed by a unique, unambiguous identifier such as a sequentially assigned integer. (A location reference method may be ambiguous due to inherent measuring and instrumentation errors).
2	Geometry and spatial characteristics	• A place may be represented by a point. Its shape may be represented by Geometry (as described by the Geographic Markup Language–GML standard) which is a variable data type that may be used to describe different type geometries (e.g., one, two and three dimensional shapes). • All features should include Latitude and Longitude in NAD ’83. • Features may include other types of location references including: - Linear (relativeLocation) - Intersection - Planar (x-coordinate; y-coordinate) UTM Zone 18 for the downstate NY region. • *Non-Generalized* location: a specific location is one that is identified for the precise place for which the feature is located. It is not used to summarize a place where multiple features may be represented.

Table 7-1: Requirements for Location Table and Related Data

#	Category	Requirements
		• *Generalized* location: often a location may be based on a low resolution digital base map or is assigned a point associated with the intersection centerline (for example, when a transit stop is associated with a timepoint). These are referred to as "generalized" points. There are several downstream applications that prefer generalized locations, for example, scheduling, macro planning tools, etc. • A non-generalized location may point to the generalized location to which it is referred. (This may be a manual process or require a tool to correlate the applicable fields.)
3	Feature Types	• The SDP includes several feature types or place categories. These include: - Transit Facility - Transit Stop - Timepoint - Transfer Cluster - Amenity, portal, passenger access component - Physical Point used in by an event or trip time Other feature types may be included by local agencies as extended attributes.
4	Configuration Management of Location versus "Place" type	• A transit place may change location (and consequently locationID), although it may not change functionality, for example, a transit stop may be moved from the near side to the far side of the intersection. If the locationID is generalized, then the relativeLocation may change, and the effectiveDate and endDate are modified. However, it the location is non-generalized, a new locationID should be defined for the stop or facility.
5	Associations	• Several entities make use of the Transit Gazetteer to build the transit physical and service networks. These include: - Trip Time - Transfer Cluster (and by association Connection Segment) - Transit Path - Transit Path Event - Transit Point Event - Block Event Time - Pattern - Trip - Transit Facility, Amenity, Portal, Passenger Access Component, Transit Stop, Depot - Timetable Header

Conceptual Data Reference Model (CDRM) Description for Transit Gazetteer

The Gazetteer Branch describes the location and places used to build the transit network. The CDRM for the Transit Gazetteer branch is shown in Figure 7-5.

The Transit Gazetteer's Location Table serves to extend and centralize transit feature location references for transit point features: Transfer Cluster, Transit Facility, Timepoint and Transit

Stop, Connection Segment. These features, once aggregated, may provide a single look up table or definition to build the transit network and locate transit facilities.

The Location Table is composed of several types of location references, spherical (latitude/longitude), planar, linear and attribute (relativeLocation). In addition, a generalized Location may be associated with several (non-generalized) Locations.

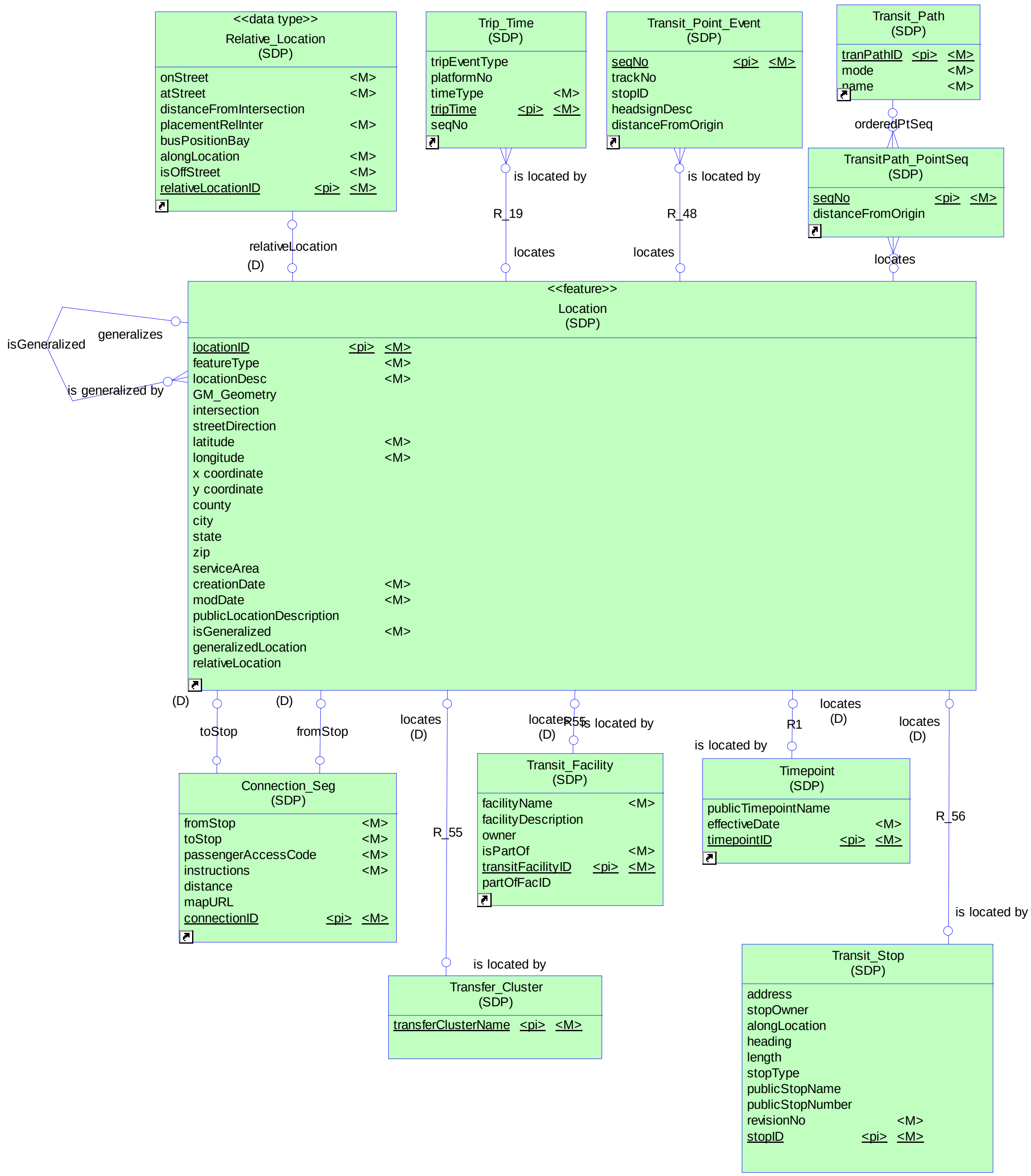


Figure 7-5: Transit Gazetteer CDRM

SDP XML Schema Description for Transit Gazetteer

In applying the CDRM to generate the XML Schema a number of rules were applied to the Gazetteer model to generate the XML Transit Gazetteer Branch.

The high level XML Schema for the Transit Gazetteer is shown in Figure 7-6. The rules associated with the Transit Gazetteer element include:

- The Location is a mandatory element that describes all the geographic places used by elements in the Transit Facilities, Transit Network, Service and Agency Registration branches.
- Related identifiers or keys are declared using the schema related key constraint: keyref; the keyref constraint was referenced to the Location element's locationID (declared as schema "key" constraint). These constraints ensure that when a locationID is used by another element, it checks to ensure the locationID exists as a record in the Location Element.

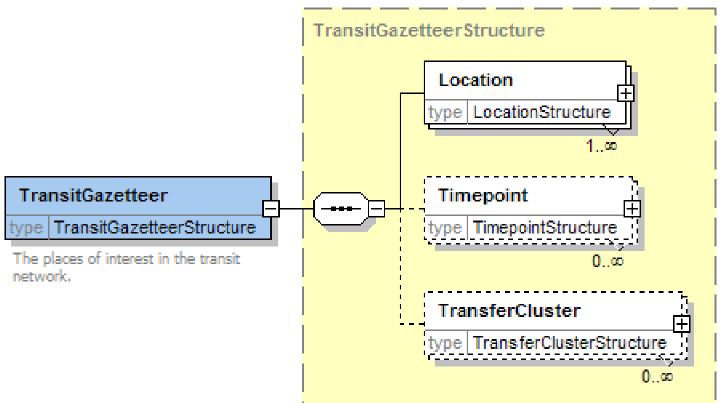


Figure 7-6: SDP XML Excerpt for Transit Gazetteer

The XML Schema for a Location element is shown in Figure 7-7. The rules applied to the Location element include:

- Seven elements are mandatory including: locationID, featureType, locationDesc, longitude, latitude, publicLocationDescription and isGeneralized
- In most cases the isGeneralized element will be "false"; use it only if there is another locationID which is pointed to it. The feature was developed to generalize multi-use facilities such as Battery Park, Grand Central Terminal and Pennsylvania Station in NYC.
- Intersection should include a "@" symbol between the "at" and "on" streets. This is used to parse the data if necessary.
- Avoid using "&" and other html type characters in the descriptions. The XML validator will declare an error.
- Geometry uses the GML description to describe the physical polygon. See Section 4.4 for an example of the polygon type.
- State, street direction and feature type are enumerated types. Currently, state only includes values NY, CT and NJ. The list should be extended as additional agencies in other States use the schema.
- Relative location references another complex element called RelativeLocationStructure
- communityName is any name which may be recognized by local users.

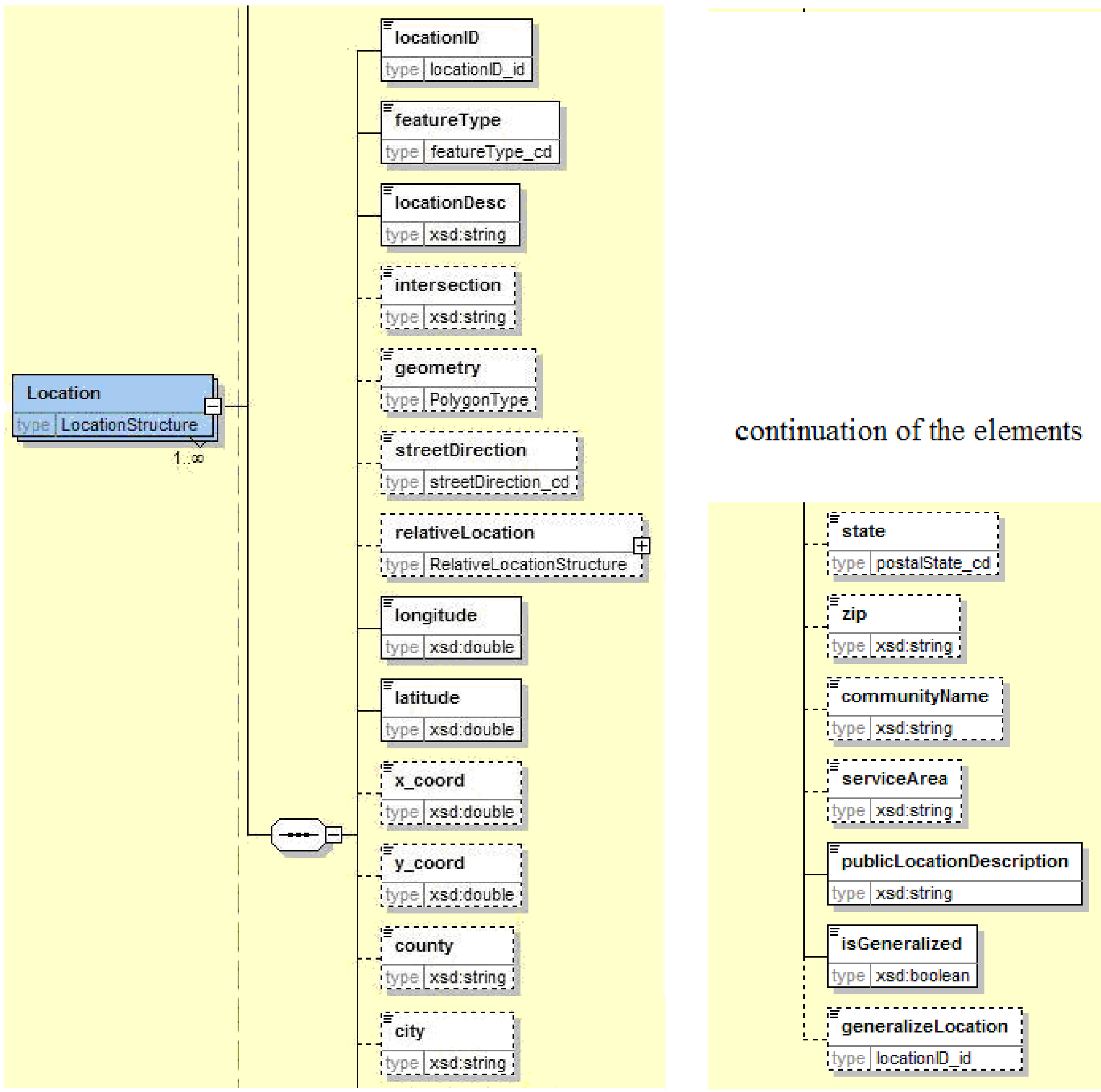


Figure 7-7: XML Schema Excerpt for Location (in two columns)

- The XML Schema for the relativeLocation element is shown in Figure 7-8. There are no additional rules applied to the relativeLocation element beyond the normal XML Schema constraints identified in the CDRM.

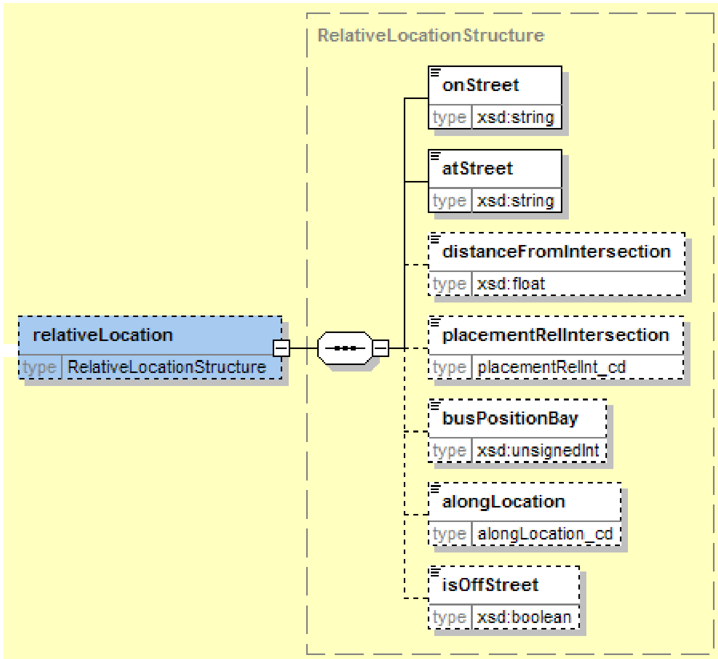


Figure 7-8: SDP XML Schema Excerpt for Relative Location

- The XML schema for the Timepoint element is shown in Figure 7-9. There are no additional rules applied to the Timepoint element beyond the normal XML Schema constraints identified in the CDRM.

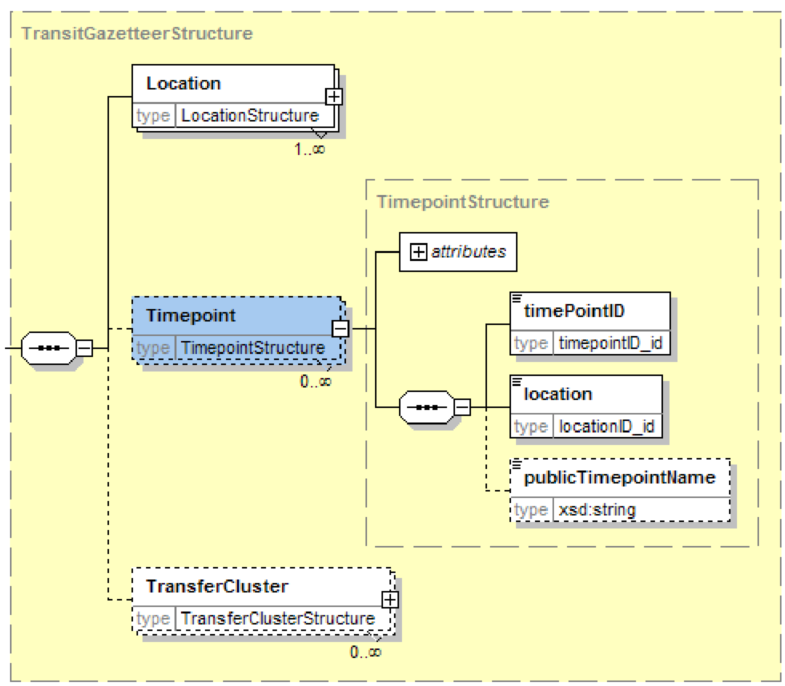


Figure 7-9: SDP XML Schema Excerpt for Timepoint

Transfer Cluster is described in more detail in Chapter 10: Advanced Topics. The details of creating transfer clusters, connections and coordinated transfers are discussed in Section 10.4.

Detailed Data Descriptions and Guidance

This section describes the best practices and guidance associated with the complex elements associated with the Transit Gazetteer, Location, RelativeLocation (embedded in Location) and Timepoint.[8] The guidance for each element is consolidated into a table with the following column headings: Requirement status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

Table 7-2: Location Formats and Guidance

Req	Element Name	Type	Questions to Ask
M	locationID	locationID_id UNIQUE	Does your data source use a location identifier like geography identifier or AVL Physical Point to designate location referencing information for a stop, timepoint or other spatial feature?
M	featureType	featureType_cd	Is there a way to tell the difference among the following transit features (other than special tables)? (e.g., stops features are alphanumeric while timepoints are only numeric). Also, transitStop value takes precedence over collocated a timepoint value.
M	locationDesc	string	You may use any description or concatenated description in this field.
O	intersection	string	The full name of an intersection. For easier parsing, the accepted convention is "on street" @ "at street." Avoid using "&" symbol.
O	geometry	gml:GM_Polygon	The geometry of the point. A point that designates a station, for example New York Penn Station, may be instantiated as a point or polygon using different map resolutions. See Section 4.4 for guidance on how to use this element.
O	streetDirection	streetDirection_cd	The general orientation of the street, north/south, east/west, north, south, east, west. North/south and east/west may be used for either direction.

[8] TransferCluster, also included in the Transit Gazetteer branch is discussed in Chapter 10.

Table 7-2: Location Formats and Guidance

Req	Element Name	Type	Questions to Ask
O	relativeLocation	RelativeLocation	If linear referencing methods are used to describe location (e.g., nearside, off street, distance from intersection) then the relativeLocation field should be completed. See Table 7-3: Relative Location and Guidance below.
M	longitude	double	This field is a floating point. You may need to convert your data from an integer. WGS 84 is the default datum. If you use another, then record datum in metadata.
M	latitude	double	This field is a floating point. You may need to convert your data from an integer. Record datum in metadata.
O	x_coord	double	Do you use Easting/Northing (UTM Zone 18) for x and y coordinates? If you do, and do not use UTM Zone 18, then record projection and zone in metadata.
O	y_coord	double	Do you use Easting/Northing (UTM Zone 18) for x and y coordinates? If you do, and do not use UTM Zone 18, then record projection and zone in metadata.
O	county	string	Standard name for county.
O	city	string	Standard name for a city.
O	state	state_cd	Populated from ANSI/FIPS codes.
O	zip	string	May be zip +4.
O	communityName	string	Local community name in which the location is known.
O	serviceArea	string	A description of the service area. May be defined as a fare zone.
M	publicLocationDescription	string	This is the field that is used to describe the location to the public. The field should not contain abbreviations or codes, and it should use proper punctuation and capitalization.
M	isGeneralized	boolean	If this location is contained in, is part of, or has a "many-to-many" relationship to another location, then isGeneralized is 'false.' For example, a timepoint location may be associated with one or more bus stop locations. In the timepoint/bus stop case, the timepoint is true and the bus stops are false.
O	generalizeLocation	locationID_id	When isGeneralized is false, and the feature points to another location, then the specific record index (locationID) to which it points is included in this field. For example, WWM01 and WWM02 point to the "WWM" locationID.
O	effectiveDate	date	[attribute]
O	endDate	date	[attribute]

Table 7-3: Relative Location Format and Guidance

	Element Name	Type	Questions to Ask
M	onStreet	string	The name of the street on which the feature is located; if the relative location is not on the street, then this is the nearest street location.
M	atStreet	string	The name of the cross street nearest the location.
O	distanceFromIntersection	float	The distance of the feature from the intersection. An attribute may be designated for either feet or meters. The default is assumed to be feet when no attribute is present.
O	placementRelIntersection	placementRelIntersection_cd	Codes include: nearside; farside; midBlock; at; between; farsideMidBlock; nearsideMidBlock; opposite.
O	busPositionBay	integer	This is a number that designates the platform, bus bay, ferry berth or other access area that may not be defined in the TransitStop element.
O	alongLocation	alongLocation_cd	The location along a boarding area where passengers board or alight a transit vehicle. Valid fields refer to the position based on the direction of travel: right, left, both.
O	isOffStreet	boolean	If the location is off the street, then this field is true; otherwise it is assumed the location is considered on the street.

Table 7-4: Timepoint Format and Guidance

	Element Name	Type	Questions to Ask
M	timepointID	timepointID_id UNIQUE	Insert the native field that designates the timepoint.
M	location	locationID_id	Insert an index, locationID, that refers to the Location element.
O	publicTimepointName	string	Is there a timepoint name or description known to the public?
O	effectiveDate	date	[attribute]
O	endDate	date	[attribute]

Usage and Examples of Transit Gazetteer

Example 1: Location Table—Need for process/data checks

The Location Table is fairly straightforward to implement. The innovative feature is the publicLocationDescription. This element is used to support key downstream applications such as timetables and trip planning applications. The NYC Bus data set includes a public location reference of intersecting street/landmark names or a landmark. Intersections are described by the "on" street (or landmark) joined to the "at" street with a "@" symbol. Typically, the publicLocationDescription is the intersection name. Sometimes the name used by the public is a landmark name like Port Authority Bus Terminal, Walt Whitman Mall or Kingsbridge Depot.

Care should be taken to check whether the native data is referencing an **intersection** like GRAND CONCOURSE @ E 170 ST or a **landmark** like KINGSBRIDGE DEPOT so that the publicLocationDescription does not read like “the landmark at the landmark,” such as: KINGSBRIDGE DEPOT @ KINGSBRIDGE DEPOT.

Description for an Intersection

```
<Location>
  <locationID>10ca</locationID>
  <featureType>transitStop</featureType>
  <locationDesc>GRAND CONCOURSE</locationDesc>
  <intersection>GRAND CONCOURSE @ E 170 ST</intersection>
  <longitude>-73.913296</longitude>
  <latitude>40.838995</latitude>
  <publicLocationDescription>GRAND CONCOURSE at E 170 ST</publicLocationDescription>
  <isGeneralized>false</isGeneralized>
</Location>
```

Description for a Landmark

```
<Location>
  <locationID>3eac</locationID>
  <featureType>transitFacility</featureType>
  <locationDesc>KINGSBRIDGE DEPOT</locationDesc>
  <intersection>KINGSBRIDGE DEPOT </intersection>
  <longitude>0</longitude>
  <latitude>0</latitude>
  <publicLocationDescription>KINGSBRIDGE DEPOT </publicLocationDescription>
  <isGeneralized>false</isGeneralized>
</Location>
```

Example 2: Representing Places with Relative Location Information

The Long Island Bus data set includes relative location information in the form of on and at streets to describe a place. In addition, the location description tag (e.g., <locationDesc>) includes references to the physical point identifier and bus stop identifier used in the native data set. Notice how the fields embedded in the relativeLocation element are indented slightly. This format is needed to provide a “well formed” SDP XML Document.

```
<Location>
  <locationID>234</locationID>
  <featureType>transitStop</featureType>
  <locationDesc>HILLSIDE AVE nearest bus stop J179 and Phys Pt J1791</locationDesc>
  <relativeLocation>
    <onStreet>HILLSIDE AVE</onStreet>
    <atStreet>179TH PL</atStreet>
  </relativeLocation>
  <longitude>-73.783251</longitude>
  <latitude>40.712695</latitude>
  <publicLocationDescription>HILLSIDE AVE at 179TH PL</publicLocationDescription>
  <isGeneralized>false</isGeneralized>
</Location>
```

Example 3: Timepoint Example

In addition to using "street names at an intersection" as a location referencing method, the NYC Bus data set also contains timepoint and bus stop identifiers. In the data conversion process, this location information may be inserted in the Location (locationID = 173f), TransitStop (stopID=300200 where locationID=173f) and Timepoint (timepointID= CIAAU where locationID=173f) elements.

```
<Location>
  <locationID>173f</locationID>
   <featureType>transitStop</featureType>
  <locationDesc>AV U at CONEY ISLAND AV</locationDesc>
  <intersection>AV U @ CONEY ISLAND AV</intersection>
  <longitude>-73.960717</longitude>
  <latitude>40.598289</latitude>
  <publicLocationDescription> AV U at CONEY ISLAND AV </publicLocationDescription>
  <isGeneralized>false</isGeneralized>
</Location>
….
<Timepoint>
  <timePointID>CIAAU</timePointID>
  <location>173f </location>
  <publicTimepointName> AV U at CONEY ISLAND AV </publicTimepointName>
</Timepoint>

<TransitStop>
  <stopID> 300200</stopID>
  <locationID>173f</locationID>
  …
  </TransitStop>
```

Chapter 8: Transit Facilities Data Concepts

In This Chapter

- Understand Transit Facilities Data Concepts and requirements.
- Understand how to represent Transit Stops in the SDP.
- Learn how to map your native data to the elements in the Transit Facilities Branch.
- Explore the scalability of the Transit Facilities Model for representing complex multimodal and/or multi-agency facilities.

Transit Facilities Branch Definition

The Transit Facilities Branch of the SDP XML Schema includes the entities associated with fixed assets owned or used by transit in the operation, delivery or maintenance of public transit services.

The model includes the following data concepts:

- Transit Facility including Plant Component and its Status
- Amenity
- Passenger Access Component
- Portal
- Transit Stop
- Track

Purpose of the Transit Facilities Branch Model

The SDP's Transit Facilities Branch model and data concepts were developed to meet the complex requirements of the transit stop in the downstate New York region. The full range of stops is covered by the model, from the simple case of single-mode, single-route bus stop, to the complex case of describing a stop in a multimodal facility like Penn Station.

The Transit Stop is "an established location where public transportation customers may board or alight a transit vehicle in revenue service." Further, the Transit Stop is an important data concept required to meet transit business requirements. However, based on a review and analysis of the New York region's schedule data requirements, the Transit Stop data concept was not sufficient by itself to capture the transit needs of a complex, multimodal, densely urban environment served by multiple transportation agencies. In the SDP, the Transit Facilities data concepts and models were developed to more effectively describe transit stops, including their locations and amenities.

What is a Transit Facility?

A Transit Facility, in its simplest case, is a bus stop or location where a passenger may board or alight a revenue vehicle. A more complex case of a Transit Facility would be a multimodal, multi-agency facility such as Penn Station. A Plant Component, such as a Transit Stop (one alias is Boarding Area), is a physical part of a larger Transit Facility. These data entities, Transit Facility and Plant Component, are further described in Table 8-1.

The most simple case of a bus stop may be described by a Plant Component whose type is boarding area. Additional amenities such as the marker/sign, shelter, bench, and schedule information may be designated as part of the Transit Facility. The simple case is illustrated in Figure 8-1.

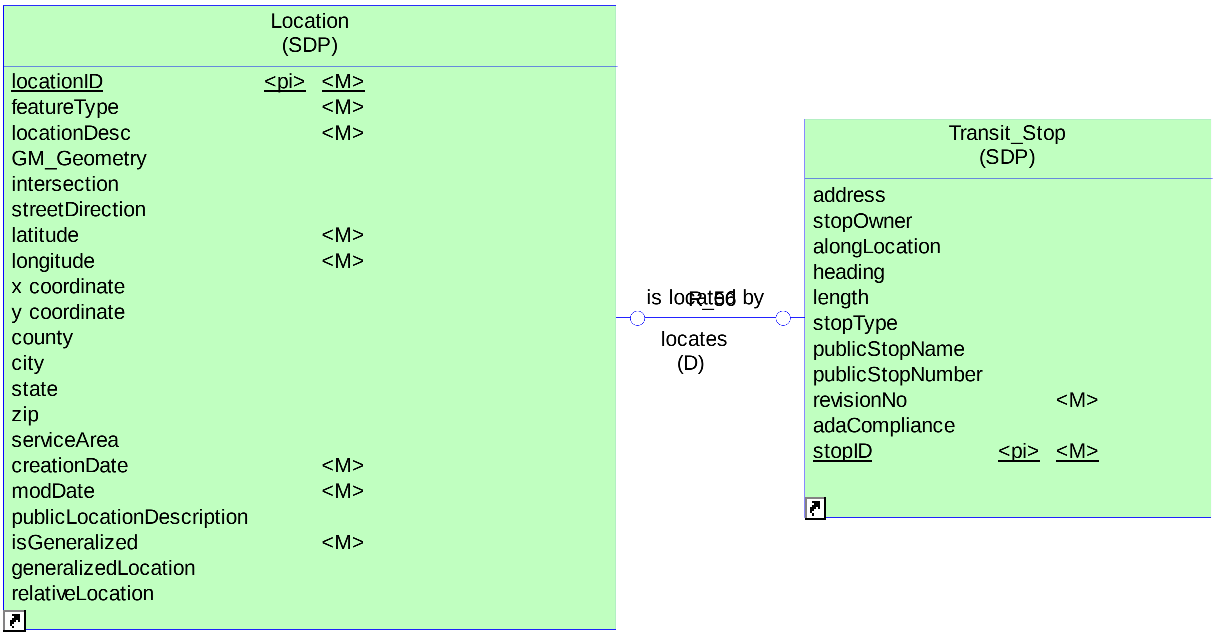


Figure 8-1: Simple Model for Transit Stop

However, rail and other multimodal services do not fit well into this simple model. A Facility may contain many Plant Components and Plant Components may be shared by many facilities. For example, the Port Authority Bus Terminal (PABT) contains facilities for Coach USA, New Jersey Transit, New York City Transit, etc. The operator's facilities may be aggregated into a larger facility called PABT. Together they include Plant Components such as entrances and exits, stairs, elevators, escalators, benches, electronic signs and other amenities. A selection of the Plant Components may serve one or more of the boarding areas or bays for New York City Bus, Coach or New Jersey Transit buses. The conceptual data reference model for the SDP, shown in Figure 8-2, addresses the needs and requirements of both the simple and complex cases.

Although the SDP is designed to handle complex transit facilities/stops, most bus operators will only need to submit a ***list of bus stops*** (i.e., Boarding Area) and related locations. A bus operator, with only simple bus stops, will only need to submit their data per the simple model for a Transit Stop as illustrated in Figure 8-1. When the Transit Schedule Data Exchange Architecture integrates the data into a regional application, it may transform the data into the more complex model for representing Transit Facility data.

Table 8-1 provides a list of definitions of the entities used in the SDP's Conceptual Data Reference Model (CDRM) for Transit Facilities. The requirements that constrain the CDRM are listed in Table 8-2.

Table 8-1: Description of Transit Facility/Stop CDRM Entities (in alphabetical order)

Entity	Description/Definition
Amenity	Elements of a physical feature, a fixed location, or a Transit Facility. Example: the amenities of a public transportation stop may include shelter, platform announcement panel, park and ride lot, fare vending equipment and benches. Note: An Amenity may be described by one or more characteristics, or attributes, such as the year of construction or its current condition. [GOS, part 7d, p., 2]
Facility Plant Component Association	An association between a Transit Facility and Plant Component. The Facility and Plant Component has a many to many relationship, that is, a Plant Component such as stairs may be part of two facilities, and there may be more than one set of stairs associated with a single facility. This association distinguishes among the many relationships.
Location	This entity represents the location defined in the Transit Gazetteer. It contains the location description for nodes used to describe or relate a transit network over which transit service is provided.
Passenger Access Component	The components used to aid travelers to traverse from one level to another or from one end of a facility to another. Examples include stairs, elevator, escalator, moving walkway. The component may be described by direction (up, down, or both), accessibility for people with disabilities or carts, and other characteristics.
Plant Component	A Plant Component is a physical part of a larger Transit Facility. such as, a boarding area, turnstile, fare vending machine, information booth, escalator, stairs, etc. The types of Plant Components included in this model are Transit Facility, Transit Stop or boarding area, Amenity, Passenger Access Component and Portal. Some examples are boarding area, turnstile, fare vending machine, information booth, escalator, stairs, etc.
Platform Track Association	An association between a specific platform and track. A Platform may be associated with multiple Tracks, for example, platform B at Jamaica Station is flanked on both sides by Tracks 2 and 3. Alternatively, a track may support multiple platforms. This entity distinguishes the combined relationship between one platform and an adjacent track.
Portal	A place where transit customers may enter or exit a Transit Facility, station or stop. Examples include doors and gates to transit facilities.
Track	"A pair of parallel rails, and required ties and fastenings, over which trains move." [LIRR] May also be instantiated as a Lane or Berth.
Transit Facility	A building or center used by a transit vehicle or transit operator for the purpose of parking, storing, maintaining or providing services to transit customers. The SDP uses this entity to represent multiple transit stops, amenities, passenger access components and/or portals.
Transit Stop	An established location where public transportation customers may board or alight a transit vehicle in revenue service. Alias: bus stop, boarding area, ramp, platform.

Requirements for Transit Facilities Data Concept

Based on the environment of the New York Metropolitan region, key requirements[9] that drive the model of a transit facility (stop), both simple and complex, include the Transit Facility/Stop is:

- May represent different types of boarding areas and their constituent parts (e.g., platforms associated with appropriate tracks)
- Unique across all types of stop, amenities, portals (entrances), access components (stairs), etc.
- Logically and physically consistent with location and relative location references
- Logically and physically consistent in its relationship to other stops and Plant Components
- May represent status (temporary, permanent and planned) of each type of Plant Component
- May share and reallocate Plant Components within and among facilities (e.g., represent a subway station as a different facility then the Long Island Rail Road platforms at Penn Station, yet share the entrances, stairs, escalators, signs, etc.)
- May embed stops or facilities within facilities (e.g., Coach and. New Jersey Transit facilities at Port Authority Bus Terminal)

These requirements associated with Transit Facility, Transit Stop and related data concepts are described and categorized in Table 8-2.

Table 8-2: Transit Facilities Requirements

#	Category	Requirements
1	Transit Facility/Stop classification categories	A Transit Facility / Stop may serve different modes, for example, • Bus/Commuter Bus • Commuter Rail/intercity rail • Subway • Ferry • Multiple modes (e.g., Grand Central Station, Port Authority Bus Terminal, Penn Station, Jamaica Station) A Transit Stop may be described by a: • single mode, single route stop • single mode, multiple route stop • multiple mode, multiple route stop A Transit Stop may be contained in a • a facility (i.e., Transit Center) that contains multiple transit stops and shares multiple portals, access equipment (e.g., stairs), and amenities. • collection of stops known by a single name

[9] Stop usage information such as boarding/alighting is associated with pattern and trip events.

Table 8-2: Transit Facilities Requirements

#	Category	Requirements
2	Geometry and spatial characteristics (stops and facilities)	• Stops may be located using many different geometric references, e.g., field measures using GPS; state plane, map geometry, linear references, and address attributes. • Stops may be located using linear references relative to the transportation network (e.g., address, relative location to intersection) as well as relative to the transit network (distance or time traveled from start of pattern or start of track). • Stops may be located on- or off-street, when off street, the location may be in a building, underground, in a mall, commercial park, parking lot, or private road. • Transit centers may be described as a facility and inherit the geometry of a facility (polygon with levels), or typically, may be represented as a point (centroid or with geometry or shape). • Convention for levels: Ground level is considered level "0." First basement is level "-1", etc. First floor is considered level "1." There are cases where the ground floor and level 1 are the same. The level number should correspond to the levels described by the signs and instructions displayed in the facility.
3	Unique identification, naming conventions, and references	• Stops may contain both internal and public names and numbers. • A separate, regional identifier will be created in order to match multiple stops to a single, unique, unambiguously defined Transit Stop or Transit Facility if it is used by many operators. • The regional transit stop identifier or name should be unique across transit agency published SDP XML Documents in a regional schedule data repository. (See Shared Facility Naming Conventions, in the SDP Functional Requirements document, Appendix B.) • An agency's Transit Stop identifiers should be unique within the Agency's authorized SDP XML Document(s). • Data Types used for names and numbers - The data types for internally used names and numbers may be constrained. - The data types for public names and number will not be constrained. • If the stop name is the name of the closest intersection, then the following convention shall be used (note capitalization): E. on Street @ N. Cross Street. Use of abbreviations for prefix, suffix, and road type are allowed as long as they follow the Addressing Conventions described by the USPS standard naming practices. • If the stop name is the same as a nearby landmark or point of interest, then the preferred spelling and abbreviation should use a regionally accessible Landmark Gazetteer. [This feature was not implemented in the demonstration project.]
4	Portal Characteristics	• Most subway stations have more then one Portal to enter and exit. • Portals may be exit only, enter only or both. • Some portals are closed during certain hours.

Table 8-2: Transit Facilities Requirements

#	Category	Requirements
5	Stops Contained within a Transfer Cluster or Transit Facility (Center)	• A Transit Facility may contain multiple transit stops within a single facility. Each stop is described as a Plant Component within the facility. Optionally, a transit stop may be described as a facility. When a transit stop is described as a Transit Facility, the stop is "part of" another Transit Facility. For example, the following stations (i.e., transit facilities) may be defined as" part of" New York Penn Station transit facility: (1) NYCT Subway Station for lines A, C, E (8th Avenue Station) (2) NYCT Subway Station for lines 1, 2, 3 (7th Avenue Station); (3) LIRR; (4) Amtrak; (5) NJ Transit; (6) New York City Buses; (7) NYCT Bus; and (8) PATH (33rd & 6th). • A transfer cluster is a collection of transit stops where transfer between stops is convenient *and* scheduled. A description of a transfer cluster is driven by internal policy considerations. (See Chapter 10 for details on the Transfer Cluster Data Concept.)
6	Stop Ownership	• The stop owner is an organization that owns and maintains the stop. • More than one Transit Agency my use a stop which they do not own; they may refer to the stop using different internal public names and numbers. The Public Name spelling and abbreviations should conform to the list of transit facilities included in the Shared Facilities Naming Conventions (see SDP Functional Requirements Document, Appendix B).
7	Stop Owner and Record Submission	• A stop record may be submitted by any organization that uses a Transit Stop using their internal stop name and number. • If the stop record is submitted by an organization other then the Stop Owner, then it may contain information on the responsible Owner for the stop.
8	Amenities and ADA accessibility	• Stops may be associated with amenities and other characteristics like lighting, benches, etc. • Transit stops and facilities have accessible and non-accessible characteristics. • Although entities are developed to capture these features, these characteristics are captured when the stop is represented as a Transit Facility through the Plant Component elements.

Conceptual Data Reference Model (CDRM) Description

The conceptual data reference model for the Transit Facilities Branch is illustrated in Figure 8-2. The CDRM is described as follows:

A Transit Facility represents any building or physical location used by a transit operator or location where the public may access fixed route transit service. A Transit Facility may contain or be contained by other Transit Facilities. For example, the Port Authority Bus Terminal contains levels that are managed by other transit operators such as Coach USA and New Jersey Transit. A Transit Facility may contain multiple Plant Components such as Transit Stops and Amenities. Transit Facilities may share Plant Components. For example, Penn Station Amtrak shares stairs with the New York City Transit subway station for A, C, and E lines.

Several types of Plant Components are allowed in the SDP: Transit Stop (boarding area), Amenity, Passenger Access Component, Portal, Track and even another Transit Facility (e.g., New Jersey Transit facility in NY Penn Station). These component types are associated to each other through the Plant Component description. They are then associated to a Transit Facility or are instantiated as a Transit Facility. In its simplest form, the Transit Facility (Plant Component/Transit Stop) may be described as a bus stop. Finally, each Plant Component includes a status on its current condition, that is, whether the Plant Component is active, planned, or closed, and the dates it was created in the database, opened, closed, and more.

Transit Stop represents a key feature; it may also be known as a boarding area or platform where transit riders board or alight from transit service. The Transit Stop may be contained in a Transit Facility that contains multiple stops, where transit riders enter, depart from or transfer between transit services. A Transit Stop may be contained in a cluster of stops that form a Transfer Cluster wherein a transit rider may change services. The Transit Stop is tied to the physical geometry (as described in Location and referenced by locationID) as well as the linear description of its environment including relativeLocation associated with the nearest intersection, alongLocation, heading and address. Service status conditions such as stopOwner are key attributes to determine usage and responsibility.

A Portal may have defining characteristics such as whether a rider may enter, exit, or pass in both directions, as well as the time the portal opens and closes.

A Passenger Access Component describes the access component, such as a set of stairs, moving walkway, elevator, escalator or other. The direction of travel, obstacle type (an enumeration of a potential obstacle to traversing the connection segment), and other characteristics are described. The Passenger Access Component must be associated with a type of Plant Component that is associated with a Transit Facility.

An Amenity is a physical feature of a fixed location or Transit Facility. Examples of amenities of a public transit stop may include the park and ride lot, shelter, platform announcement panel, and benches. An Amenity must be associated with a type of Plant Component that is associated with a Transit Facility.

A track and platform may have a many to many relationship. The Platform Track entity distinguishes which platform (e.g., Transit Stop) is associated with the track served.

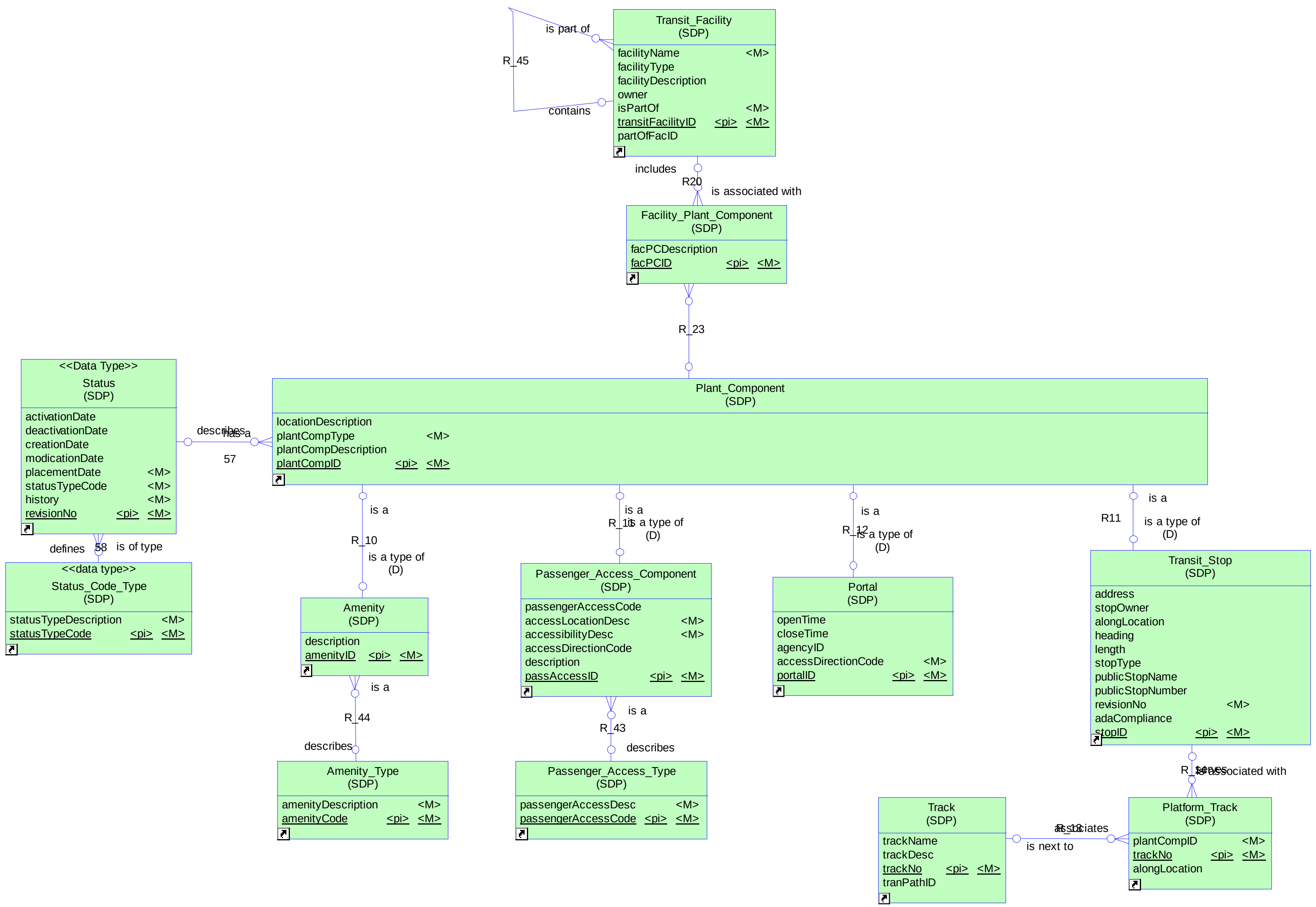


Figure 8-2: Transit Facilities CDRM

High Level SDP XML Schema Model Description for the Transit Facilities Branch

A number of rules were applied to the CDRM to generate the Transit Facility segment of the SDP XML Schema. These included:

- Each entity of the Transit Facilities model was included as a separate element.
- Associated elements, such as Plant Components associated with a Transit Facility, or Tracks associated with a Transit Stop, are embedded in the parent element.
- Most major elements branching off TransitFacilities (i.e., Amenity, Portal, TransitStop, PassengerAccessComponent) also includes a locationID in the event that the "asset" is not contained in a TransitFacility as a plantComponent.

The SDP XML Schema fragment for the high level TransitFacilities branch is illustrated in Figure 8-3.

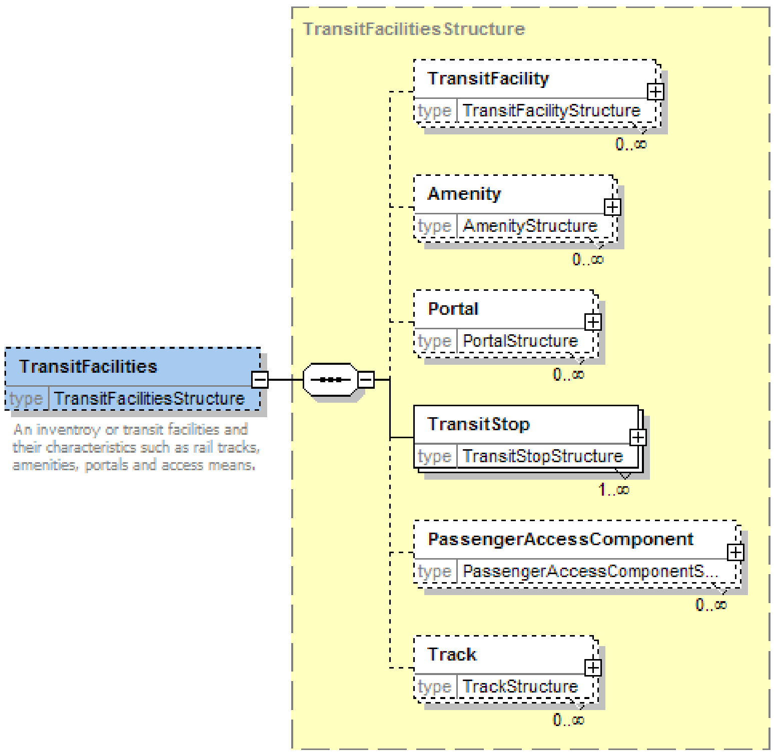


Figure 8-3: High Level SDP XML Schema Fragment for the Transit Facilities Branch

Transit Facility: SDP XML Schema Fragment

In applying the Transit_Facility entity from the CDRM to generate the SDP XML Schema's TransitFacility element illustrated in Figure 8-4, a number of rules were applied. These include:

- isPartof is a Boolean type that when true, the facility is part of another facility, such as Long Island Bus Mineola Bus Terminal in the Mineola Intermodal Transit Center.
- partOf indicates the transitFacID of which the Transit Facility is part. In the example above, partOf element would contain the transitFacID for Mineola Intermodal Transit Center. There are several "authoritative" names for shared facilities like the Mineola

Intermodal Transit Center. The list of Regional Shared Facility names is included in the SDP Functional Requirements Document, Appendix B.

- plantComponentList is an optional element that contains the list of components contained in the facility. The definition of the Plant Component structure is described below.
- facilityDescription may contain additional characteristics of the facility either in a structured/tagged format or open values.

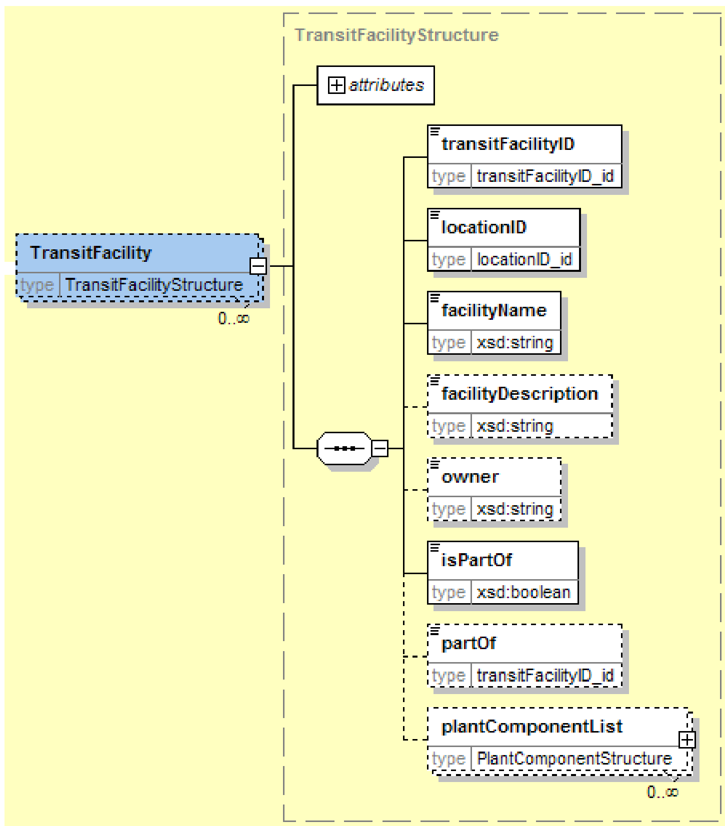


Figure 8-4: SDP XML Schema Fragment for the Transit Facility Element

Plant Component: SDP XML Schema Fragment

In generating the Plant_Component CDRM entity into the SDP XML Schema fragment illustrated in Figure 8-5, a number of rules were applied. The rules include the following:

- plantCompID should be unique within the plantComponentList.
- componentID inherits the identify of the Plant Component it represents, an Amenity will include the amenityID in the Amenity inventory list, a Transit Stop will include the stopID of the TransitStop inventory list.
- plantCompType defines the type of Plant Component. It can be characterized as one of seven types of Plant Component: Amenity, Portal, Passenger Access Component, Transit Facility, Track and Transit Stop.
- locationDescription is a description of the Plant Component in the context of the Transit Facility. Since the region has not yet identified a simple automated way of describing location within a Transit Facility, an expository description should be used.
- Similar to facilityDescription, plantCompDescription is an open format where either a description or structured tags may be stored.

- Each Plant Component should optionally include a status element that describes the operational usage and availability of the component within the Facility.
- The Plant Component does not include a locationID because it inherits the location of its parent TransitFacility.

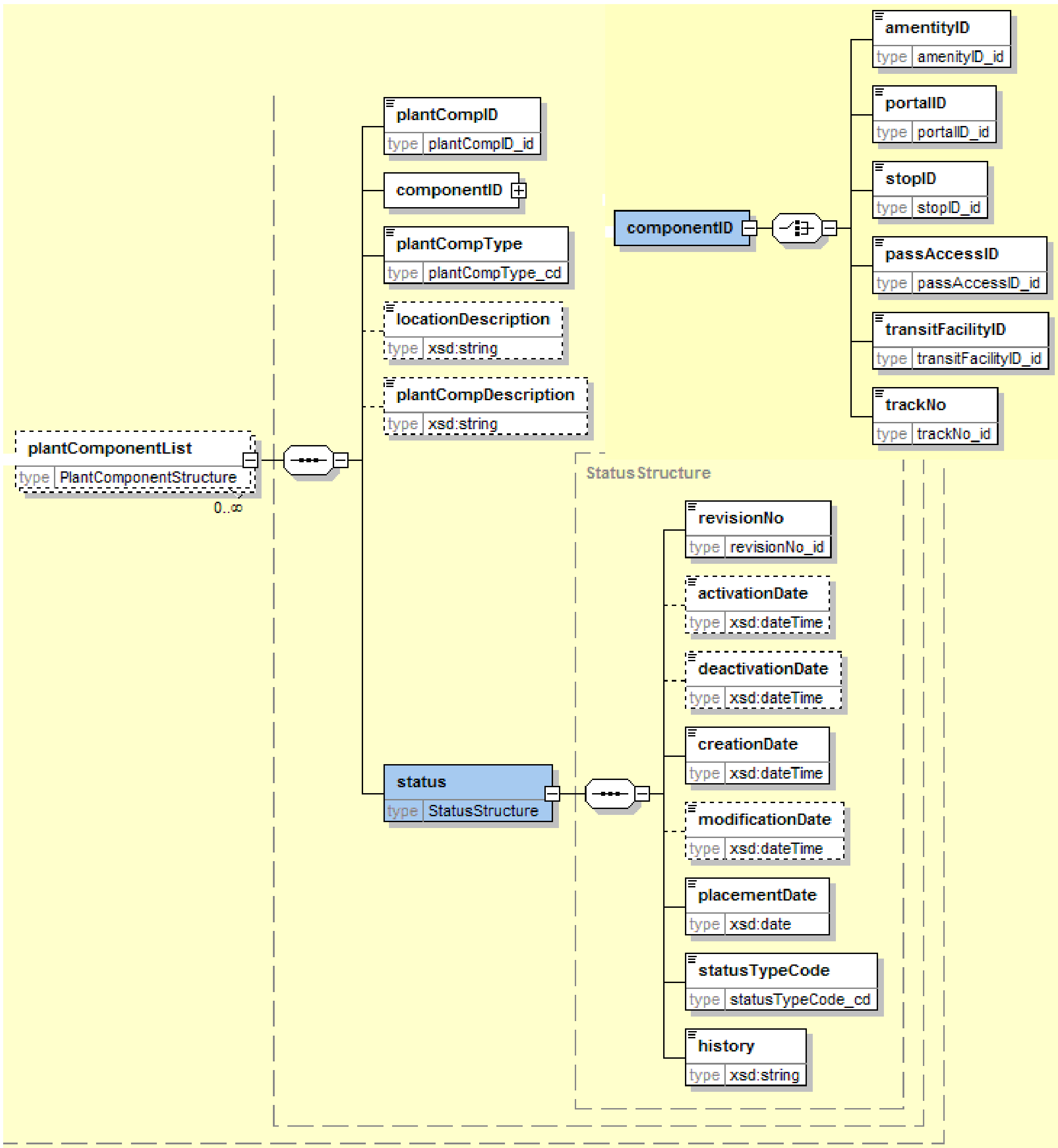


Figure 8-5: SDP XML Schema Fragment for the Plant Component Element

Amenity: SDP XML Schema Fragment

In generating the SDP XML Schema fragment shown in Figure 8-6 for Amenity, from the CDRM's Amenity entity, a number of rules were applied. These rules include the following:

- locationID is included as an optional related element so instances of Amenity may stand as an independent inventory of transit amenities.
- amenityCode was migrated to an enumerated value.
- description is an open value and may be used in an open form or as a structured (tagged) list of additional Amenity characteristics. For example, sidewalk may include additional attributes such as: Sidewalk Surface. Sidewalk Width, Sidewalk Condition, Curb Cuts, Adjacent Land Use [attributes from Bee Line bus stop data set].
- attributes effectiveDate and endDate were included for data configuration management purposes.

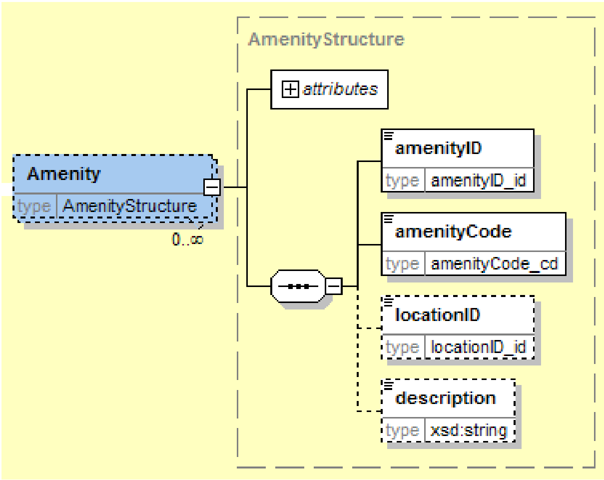


Figure 8-6: SDP XML Schema Fragment for the Amenity Element

Portal: SDP XML Schema Fragment

In generating the SDP XML Schema fragment shown in Figure 8-7 for Portal, from the CDRM's Portal entity, a number of rules were applied. These include the following:

- attributes, effectiveDate and endDate were included for data configuration management purposes.
- locationID is included as an optional related element so instances of Portal may stand as an independent inventory of entrances/exits to transit facilities.

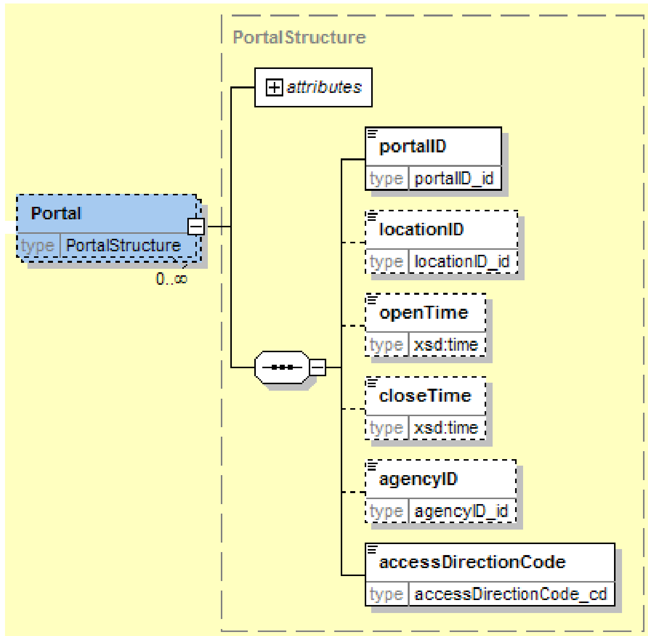


Figure 8-7: SDP XML Schema Fragment for the Portal Element

Transit Stop: SDP XML Schema Fragment

In generating the SDP XML Schema element shown in Figure 8-8 for Transit Stop, from the CDRM's Transit_Stop entity, a number of rules were applied. These include the following:

- stopID should be unique for all stops described by the Agency. There may be special considerations for rail systems that use this data concept to describe a platform. (See Appendix A on Special Considerations for Rail).
- locationID is included as an optional related element so instances of TransitStop may stand as an independent inventory of stops.
- AddressStructure describes the individual elements of the stop address. It is described in Part 3 SDP XML Schema Template.
- adaCompliance, alongLocation and stopType are described as enumerated values.
- length is a floating point type with an attribute that identifies the unit as either feet or meters. Feet is the default value if an attribute is not present.
- heading is in decimal degrees.
- publicStopName and publicStopNumber should be documented for public presentation.
- trackList embeds the track, bay or dock that is associated with the transit stop, platform, or more generally, the boarding area. There may be a many to many relationship among stops (platform) and tracks.
- Attributes, which include effectiveDate and endDate, are intended for data configuration management purposes.

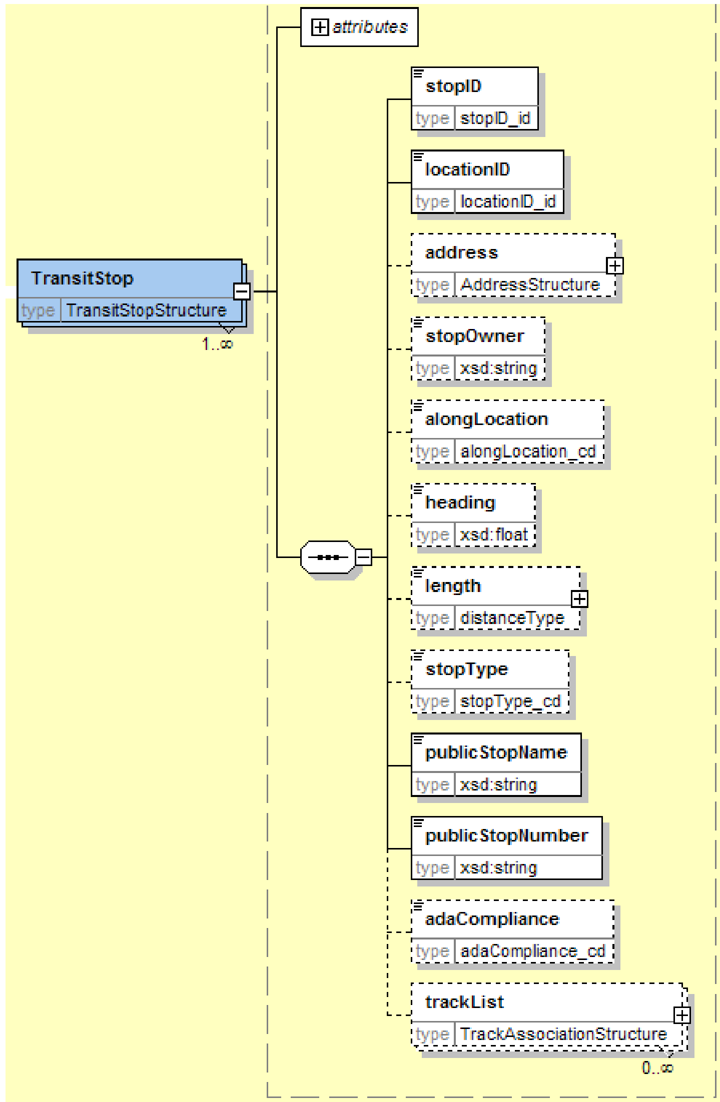


Figure 8-8: SDP XML Schema Fragment for the Transit Stop Element

Passenger Access Component: SDP XML Schema Fragment

In generating the SDP XML Schema element for the Passenger Access Component from the CDRM Passenger_Access_Component entity, as illustrated in Figure 8-9, a number of rules were applied. These include the following:

- passAccessID should be a unique value among all PassengerAccessComponents described by an Agency.
- passengerAccessCode and accessDirectionCode are enumerated type values.
- accessLocationDesc is used to describe the location of the Passenger Access Component, for example, "enter through street doors to veranda with stairs that descend to hallway."
- accessibilityDesc describes the constraints on mobility, for example, "elevator includes Braille labels."
- description is an open value and may be used in an open form or as a structured (tagged) list of additional Amenity characteristics.

- Attributes, which included effectiveDate and endDate, were included for data configuration management purposes

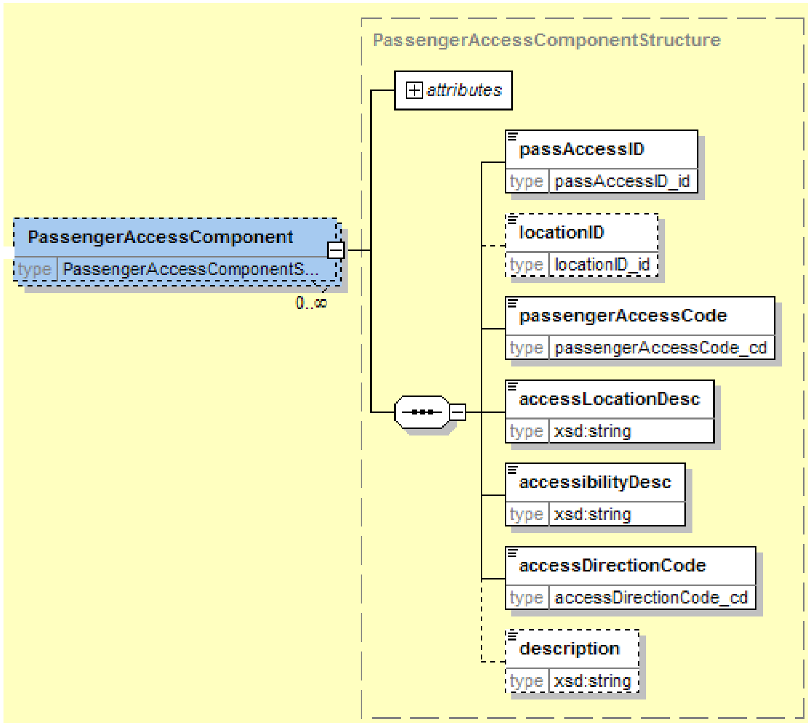


Figure 8-9: SDP XML Schema Fragment for Passenger Access Component Element

Track: SDP XML Schema Fragment

In generating the Track SDP XML Schema element from the CDRM, as illustrated in Figure 8-10, a number of rules were applied. These include the following:

- tranPathID references a Transit Path element if it was developed.
- Attributes, which included effectiveDate and endDate, were included for data configuration management purposes.

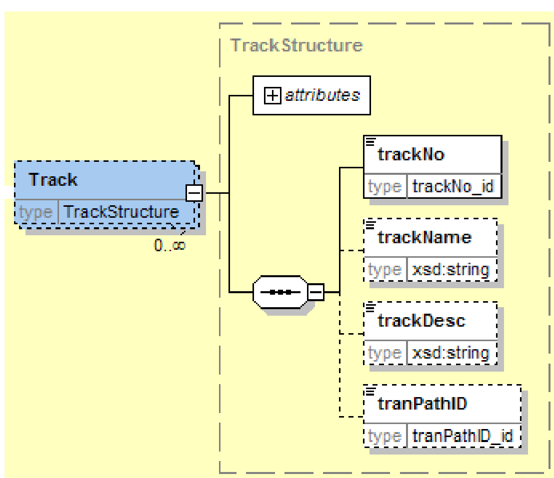


Figure 8-10: SDP XML Schema Fragment for the Track Element

Detailed Data Descriptions and Guidance for Transit Facilities

This section describes the formats and guidance associated with each high level Transit Facilities element in the data concepts described above. The guidance is consolidated into tables that include a list of baseline requirements (M for mandatory and O for optional), the element name, the data type and guidance related to element. The guidance provides clarity to the data definition. The first column of each table designates the baseline requirements based on the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet its data needs. The element name corresponds to the related CDRM entities and attributes. The type may refer to a native XML type, or declared type in the XML schema. The Guidance column is called "Questions to Ask/Guidance." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

The following tables are included in this section:
- Table 8-3: Transit Facility Guidance
- Table 8-4: Plant Component Guidance
- Table 8-5: Status Guidance
- Table 8-6: Amenity Guidance
- Table 8-7: Portal Guidance
- Table8- 8: Transit Stop Guidance
- Table 8- 9: Track Association Guidance
- Table 8-10: Passenger Access Component Guidance
- Table 8-11: Track Guidance

Table 8-3: Transit Facility Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	transitFacilityID	IDENTIFIER UNIQUE	Is there an identifier that may be used for the ID? If Transit Facility is a simple bus stop, you may duplicate stopID in this field.
M	locationID	locationID_id	A reference to the location reference element.
M	facilityName	String	The name of the Transit Facility that is known to the public. Use shared facility name if this is a description of a shared facility.
O	facilityType	facilityType_cd	A type of Transit Facility. Enumerated type. Types include: depot/garage, passenger (for boarding/alighting transit vehicle in revenue service).
O	facilityDescription	String	A description of the Transit Facility. May insert free form text or structured tag list.
O	owner	String	Insert the owner of the Transit Facility.
M	isPartOf	Boolean	Insert TRUE if this Transit Facility is part of a larger Transit Facility such as LIRR Mineola station is part of the Mineola Intermodal Facility. Otherwise, insert a value of FALSE.

Table 8-3: Transit Facility Guidance

Req	Element Name	Type	Questions to Ask/Guidance
O	partOf	transitFacilityID_id	If the Transit Facility is a facility shared by multiple agencies, use the regional shared facility ID. If it is designated as a multi-use/mode facility, but only one organization uses it, then include the transitFacilityID to which this facility is "part of."
O	plantComponentList	list of Plant Component*	The Plant Components that are contained within or associated with the TransitFacility. The Plant Components may include Transit Stops, Amenities, Portals Passenger Access Components Tracks and other Transit Facilities.
O	effectiveDate	date	[attribute] The effective date of the record.
O	endDate	date	[attribute] The end date of the record.

Table 8-4: Plant Component Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	plantCompID	IDENTIFIER UNIQUE	A unique index within each Transit Facility that distinguishes Plant Components.
M	componentID	choice of amenityID_id, portalID_id, stopID_id, passAccessID_id, transitFacilityID_id, trackNo_id	Currently there are only six types of Plant Components that may be incorporated into a Transit Facility. See XML Schema for how to document the XML format for this identifier.
M	plantCompType	plantCompType_cd	This field identifies the type of Plant Component. Only one of the six types (listed in componentID) is allowed.
O	locationDescription	string	A description of the location where the Plant Component is located. The value may be used to help passengers navigate within the facility. For example, if the Plant Component is a platform (designated as a TransitStop), then the location description may contain: "The platform is accessed via the second set of stairs from the elevators."
O	plantCompDescription	string	A description of the Plant Component. May insert free form text or structured tag list.
O	status	Status	This structure describes the deployment status of the Plant Component within the facility. See Status Guidance in Table 8-5 below.

Table 8-5: Status Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	revisionNo	revisionNo_id	This is a different revision number than the schedule revision. It is used to designate the status of the facility Plant Component.
O	activationDate	datetime	The date the Plant Component was activated (placed into service).
O	deactivationDate	datetime	The date the Plant Component was decommissioned, closed, removed or deactivated.
M	creationDate	datetime	The date this record was created.
O	modificationDate	datetime	The last date this record was modified.
M	placementDate	date	The date the Plant Component was planned or placed (not necessarily into service).
M	statusTypeCode	statusTypeCode_cd	The status of the Plant Component. Valid types include: primary, revised, temporary, obsolete, suspended or planned.
O	history	string	The history of the changes to this record.

Table 8-6: Amenity Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	amenityID	amenityID_id UNIQUE	A unique designator for any type of Amenity that may be found at a Transit Facility (not supported by other elements). The amenity list may be used as a template for Plant Component, to describe types of amenities supported by an Agency. For example, if an Agency has inventory of three types of benches, three templates may be described for the types of benches, and the Plant Component would reference the amenityID and describe the instance of where the bench type was located, as well as its Status.
M	amenityCode	amenityCode_cd	A long list of values exist to describe the specific Amenity. These may be found in the SDP_Domain.xsd schema file or in Part 3 SDP Guidance Template code list page.
O	locationID	locationID_id	This field may be redundant with the Transit Facility location, and thus should only be used for large or separated Amenity types such as parking, etc.
O	description	string	A description of the Amenity. Feel free to partition this description into structured parts.
O	effectiveDate	date	[attribute] The date the record was inserted.
O	endDate	date	[attribute] The date the record was deactivated.

Table 8-7: Portal Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	portalID	portalID_id UNIQUE	A unique designator for an entrance or exit from a Transit Facility.
O	locationID	locationID_id	This field may be redundant with the Transit Facility location. For large facilities like Fulton Station or Penn Station, this locationID may point to a specific street corner where the Portal is located.
O	openTime	time	This is the time the entrance is unlocked and may be accessed by the public.
O	closeTime	time	This is the time the entrance/exit is locked and is not accessible by the public.
O	agencyID	string	This is the agency that is responsible for the Portal. The assumption is that this is not the agency submitting the SDP Document. The registration agency identifier should be used for this field.
M	accessDirectionCode	accessDirectionCode_cd	The direction access of the Portal. 0= both in and out; 1= enter only; 2= exit only; 3=other.
O	effectiveDate	date	[attribute] The date the record was inserted.
O	endDate	date	[attribute] The date the record was deactivated.

Table 8-8: Transit Stop Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	stopID	stopID_id UNIQUE	A unique Transit Stop identifier for the system. In the case of rail, a generic Platform A, B, etc. may be used to designate the Transit Stop.
O	locationID	locationID_id	This field is required for bus stops except where a Transit Stop is associated with a Facility. Then the stop may inherit its location from the Facility.
O	address	ADDRESS	A complete address as described by the AddressStructure.
O	stopOwner	string	The owner of the Transit Stop. Typically, this may refer to the owner of the post that holds the marker, or it may refer to the surface where boarding and alighting occur, e.g., city.
O	alongLocation	alongLocation_cd	The location along a boarding area where passengers board or alight the transit vehicle. Code values include: left, right or both.
O	heading	float	The direction the transit vehicle (e.g., bus) is facing when it is boarding/alighting passengers at a Transit Stop. This field is important for real time systems. The units are in decimal degrees.
O	length	distanceType_cd	The length of the Transit Stop. This field includes an optional attribute that designates unit type. The assumed unit type is feet.

Table 8-8: Transit Stop Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	stopType	stopType_cd	The type of Transit Stop typically based on the mode of service provided. See enumeration types in the Part 3 SDP Guidance Template code list.
M	publicStopName	string	The Transit Stop name that is known to the public. The field should be populated so that it may be shown to the public as-is.
M	publicStopNumber	string	The Transit Stop number that is known to the public. This may be an index that is used to in a 511-interactive voice response telephone system.
O	adaCompliance	adaCompliance_cd	Describe the characteristics of the Transit Stop with respect to services related to specific mobility, visual and auditory challenges. Values include: ▪ notCompliance, ▪ fullyCompliant, ▪ mobilityChallengedAccess, ▪ visuallyImpairedAccess, ▪ hearingImpairedAccess, ▪ mobility-VisuallyImpairedAccess, and ▪ visually-HearingImpairedAccess, and ▪ mobility-HearingImpairedAccess.
O	trackList	list of TrackAssociation	This field is specifically for rail operators. It contains one or two references to track numbers that are associated with a platform.
O	effectiveDate	date	[attribute] The date the record was inserted.
O	endDate	date	[attribute] The date the record was deactivated.

Table 8-9: Track Association Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	trackNo	trackNo_id	The Track number that is associated with a Platform (from TransitStop.trackList).
M	alongLocation	alongLocation_cd	The side by which passengers board the train, e.g, the right side of the train or the left side of the train.

Table 8-10: Passenger Access Component Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	passAccessID	passAccessID_id UNIQUE	A unique index that designates the accessibilty of a conveyance in a Transit Facility. This element describes elevators, escalators, stairs, etc.
O	locationID	locationID_id	This field may be redundant with the Transit Facility location, and thus should only be used when separated from the facility, e.g., subway station portals.
M	passengerAccessCode	passengerAccessCode_cd	The type of obstacle that might be encountered in a Transit Facility such as: stairs, elevator, escalator, etc.

Table 8-10: Passenger Access Component Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	accessLocationDesc	string	A description of the access type, for example: "escalator in GCT from Mezzanine to Lobby floor."
M	accessiblityDesc	string	General description information of the Passenger Access Component. This may be a free form text or structure/tagged list.
M	accessDirection	accessDirectionCode_cd	The direction provided by the Passenger Access Component. The direction is based on the type of device. Moving walkways may move in an "inbound" and "outbound" direction, for example.
O	description	string	Additional description information of the Passenger Access Component. This may be a free form text or structure/tagged list.
O	effectiveDate	date	[attribute] The date the record was inserted.
O	endDate	date	[attribute] The date the record was deactivated.

Table 8-11: Track Guidance

Req	Element Name	Type	Questions to Ask/Guidance
M	trackNo	trackNo_id UNIQUE	The inventory of tracks near or at transit facilities.
O	trackName	string	The Track name if it has one.
O	trackDesc	string	A Track description.
O	tranPathID	transPathID_id	If a modal path (that is TransitPath in TransitNetwork) is defined for this Track, the reference to the TransitPath.
O	effectiveDate	date	[attribute] The date the record was inserted.
O	endDate	date	[attribute] The date the record was deactivated.

Usage and Examples of Transit Facilities

Since the transit operators in the New York region have many different configurations of transit stops and facilities, a range of examples are included on how to represent them in a SDP XML Document. The examples describe how transit facilities of varying levels of complexity may be represented using the SDP XML Schema. The description of each example is listed in Table 8-12. Each example includes one or more approaches for representing the Transit Facility. References to the Appendices in which the examples may be found are included in the last column of Table 8-12.

Table 8-12: Examples of Transit Facilities

Case	Description	Example	Comments	Appendix Reference (See Appendix E for Section #s)
1	Single transit stop on street	Location and Transit Stop entities only (Westchester Bee-Line StopID 39)	Single route	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 1-2 Bee-line.doc
2	Single transit stop on the street with amenities	Location and Transit Stop defined as a Transit Facility with amenities and Passenger Access Components (Westchester Bee-Line StopID 39)	Single route with amenities	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 1-2 Bee-line.doc
2	Single transit stop off street	SCT @ Walt Whitman Mall	Multiple routes	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 2 WWM_110106.doc
3	Single ferry berth	Haverstraw Ferry--Ossining Ferry	MNR	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_TransitStop Example 3 Haverstraw(110206).doc
4	Multiple stops (routes) at a single mode transit center/facility (not subway)	Mineola	LIB	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 4-6 Mineola_103106.doc
5	Subway station with multiple entrances/exits (single platform/multiple tracks)	96th (123) as simple case Fulton/Nassau as complex case	---	none
6	Rail Station: platform(s) and bus stop(s)	Mineola	LIRR & LIB	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 4-6 Mineola_103106.doc
7	Complex Transit Facility (multiple routes, modes and operators) – identify only facility	Penn Station with three smaller facilities	Overview, details in Cases 9-11	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 7 Penn.doc

Table 8-12: Examples of Transit Facilities

Case	Description	Example	Comments	Appendix Reference (See Appendix E for Section #s)
8	Station with platforms on both sides of a track where passengers may enter on either side	Jamaica Station	LIRR	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 8 Jamaica_v0.1a.doc
9	Transit platform in transit center	Platforms A-C/tracks 1-5 at Jamaica Station	Jamaica Station	See NYSDOT_WP_TransitFacility Examples 1-6_8_v1.0.zip NYSDOT_WP_Transit-Stop Example 9_v0.1a.doc
10	Transit Stop on street at Transit Center/Facility	Penn Station M16	New York Bus	Similar to example 4
11	Transit Platform in Station with two tracks on either side of the platform	LIRR @ Penn Station		Similar to examples 8 and 9

Chapter 9: Schedule Calendar Date, Versioning and Day Type Issues

In This Chapter

- Understand the Schedule Calendar Date Data Concept.
- Learn how to apply the elements in the Schedule Calendar Date data concept.
- Learn the business rules related to managing Version, Day Type with the Schedule Calendar Date data concept.

Schedule Calendar Date Definition

The calendar date on which a transit agency operates based on an assigned type of service (within its schedule day).

Typical Practice in Transit Using a Schedule Calendar Date

The Schedule Calendar Date is a data concept that has been adopted by only a few agencies across the U.S., although it could become very helpful to various types of downstream applications that use schedule data. Specifically, the Schedule Calendar Date concept is a Calendar where specific service is assigned to dates of operation. The assignment is accomplished through the use of a *key* assigned to each trip and included in one or more dates in the Schedule Calendar. The *key*, called a day type (also called a service type), may be used to aggregate all the trips that are associated with a specific date (e.g., New Year's Eve), or type of day (e.g., weekday, holiday, Friday). To this end, the definition of the Day Type code is critical to this data concept.

The Schedule Calendar Date concept provides a formal structure to assign the provision of service (based on version, revision, route-depot version and day type) within a single schedule version to each valid schedule date. Thus, Schedule Calendar Date format is used as a registration table to index the appropriate service that is assigned to operate on particular days of the week and to identify service exceptions on particular dates. The concept associates multiple elements together. The business rules related to building these associations are biased by policy and institutional approaches. In order for the data concept to work consistently throughout the region, several issues and best practices are recommended.

The Schedule Calendar Date works by describing the operating service by the schedule version and appropriate day type. The schedule versioning requirements are captured by the schedule version data concept. The day type requirements are typically captured by a few standard day type definitions which are codified in an enumerated type. Complexity, ambiguity, and inconsistency emerge when applying these concepts. Specific areas where these inconsistencies show up include:

- How organizational structures generate service,
- Use of schedule notes,
- Definition of a schedule day,
- Various ways of using and assigning day type codes to the service provision.

Complexity and inconsistency may arise in the following situations:

- Various types of service (day types) may be associated with different organizational units, agency defined zones or time periods on a single calendar date. How is that managed by the Schedule Calendar Date concept?

- Some agencies identify exception services in a schedule note which may not be automated or "machine-readable" by a downstream application.
- NYCT starts its schedule day at 10 p.m. the calendar day before and ends its schedule day at 3 a.m. on the following calendar day. When does the day start?
- Most service or day type codes will point to typical service on a weekday, Saturday, Sunday or holiday. This approach will work for most days of the year, but will not work for exceptions. When an agency like Long Island Bus or New York City Transit applies a Sunday schedule to a holiday, like New Year's Day, this approach falls apart. A Sunday schedule, when applied to a different day of the week must be associated with the specific date to which it applies.
- How should a day type code be assigned to extra trips that are added to busy travel dates like Labor Day, New Year's Eve and days the Mets play at Shea stadium?

These issues and recommendations for implementation will be discussed in this chapter.

Requirements for Schedule Calendar Date Data Concept

The underlying requirements that drive the Schedule Calendar Date data concept are listed in Table 9-1.

Table 9-1: Schedule Calendar Date Requirements

#	Category	Requirements
1	Agency Designated Calendars	• Each agency has service (for specific routes, routes associated originating from specific depots) associated with a calendar date. • An agency need not submit a Schedule Calendar Date on a date when no service is offered (e.g., no Sunday service). • A calendar date is associated with an Agency's schedule date even if the schedule day starts the day before or ends the day after the designated date. • Alternatively, if there are no special service days or exceptions to the standard service provisions, an agency may designate service by *standard day types*, and the regional repository will populate the calendar based on dates associated with the day types (e.g., Monday Only will cover all Mondays except for holidays that fall on Monday). • Only one set of trips for a route are operated on a calendar day. The set of trips associated with a route is distinguished by the Day Type valid during that schedule period. The route may be distinguished by the organizational unit or the depot associated with the route.
2	Unique Dates	• A Service Calendar Date is associated with a unique date in the calendar, route-depot version, revision number and schedule version identifier. Together each record shall be unique. • The calendar date must fall within the valid schedule version period with which it is associated.
3	Day Type	• The Day Type qualifies the route's trip service provision designated to operate on a specific date. • Each Schedule Calendar Date instance will be associated with one and only one day type. The day type should be described by one of the code enumerations from the data type Day Type.

Table 9-1: Schedule Calendar Date Requirements

#	Category	Requirements
		• The standard set of day types are only those described in the day type code set. • Day types should not overlap, as such, a set of business rules related to assigning service to day types should be adopted. (See business rules in *Use and Examples* section.) • *Agencies that need to describe "special days" may need to define those day types in the Day Type* XML element (in the AgencyRegistration branch). A special day type shall be described as a record by the submitting Agency in the Day Type entity.
4	Specialization of Schedule Calendar Date	• The Schedule Calendar Date may be described for any type of schedule version, revision or update.

Conceptual Data Reference Model (CDRM) Description for the Schedule Calendar Date

The Schedule Calendar Date requirements may be described in the CDRM depicted in Figure 9-1. The Schedule Calendar Date data concept in Figure 9-1 is described as:

> Transit service is scheduled for each day of operation. Service components may be scheduled to operate on different dates depending on a number of factors. These factors may be schedule based; for example, special trips are designated when there is an event at Shea Stadium or service to evacuate workers from the city during a snow storm. The Schedule Calendar Date associates the relevant schedule components (designated by the Route Depot Version, Revision, and Schedule Version) and an index related to the appropriate trips (designated by the day type) into a table which is used as a reference.
>
> A Schedule Calendar Date is created for each set of schedule version components and the trips that operate on the specific dayType. In some cases, the schedule version components are scheduled for only part of a day, for example, the schedule components vary when the Mets play games that begin at 5 p.m. versus at 7 p.m."

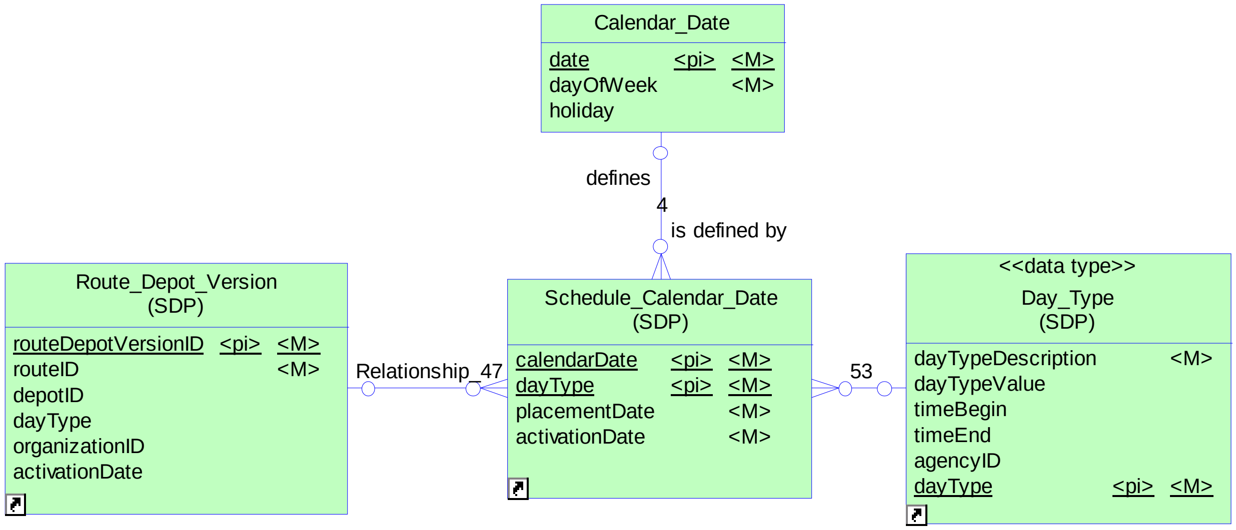


Figure 9-1: Schedule Calendar Date Concept Model

Note: The logical model will include primary and foreign keys—scheduleVersionID, revisionNo and routeDepotVersion—in the Schedule_Calendar_Date entity.

SDP XML Schema Description for Schedule Calendar Date

In implementing the CDRM into the XML Schema, a number of rules were applied.

- The Schedule Calendar Date (SCD) was generated as an independent XML Schema (called the SCD Schema), separate from the SDP. This was done in order to use the Schedule Calendar Date concept as a mechanism to integrate SDP documents, partial updates and revisions.

Figure 9-2 shows the organization of the SCD (Schedule Calendar Date) XML schema.

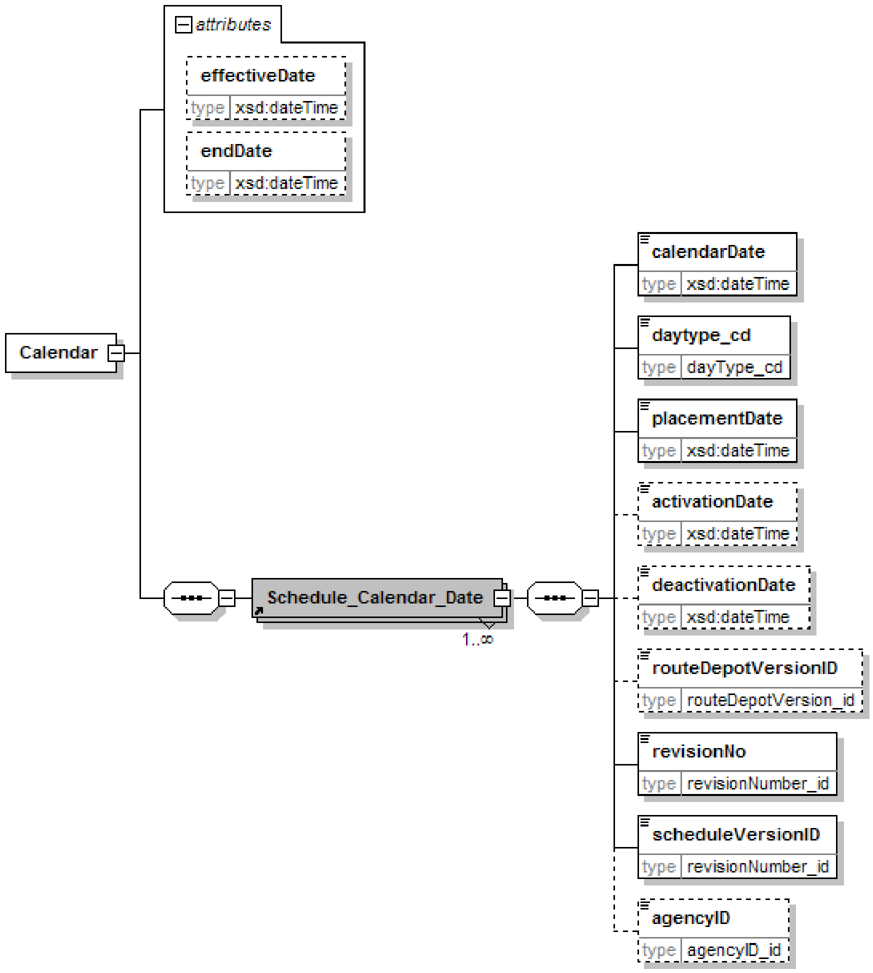


Figure 9-2: Schedule Calendar Date XML Schema Model

Detailed Data Descriptions and Guidance for the Schedule Calendar Date

This section describes best practices and guidance associated with each element in the Schedule Calendar Date data concept described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0.

A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the SDP XML Document deployment.

Table 9-2 incorporates Schedule Calendar Date Guidance.

Table 9-2: Schedule Calendar Date Guidance

Required	Element Name	Type	Questions to Ask
M	calendarDate	date	More than one calendar may be associated with different route-depot versions of the schedule. For that reason, more than one calendar date may be included in the set of ScheduleCalendarDates.
M	dayType	dayType_cd	There are a standard set of codes defined for dayType. If an agency has a need to include additional codes, a DayType element which includes a code designation and description may be developed and used as a look up table.
M	placementDate	date	The date on which this record was placed.
O	activationDate	date	The date on which this calendar date is valid. This may seem redundant with the date, however, the calendar date may be placed several months earlier, to be active during the schedule version. Later, the version may be overwritten, and the record is obsolete. This activation date should match the schedule, revision or route depot activation date to ensure consistency.
O	deactivationDate	date	See activationDate comment.
O	routeDepotVersion	routeDepotVersion_id	This element is part of the identifying key of the schedule calendar date element. The Route Depot Version designates a set of routes generated or operated by an organizational unit or depot. A unique set of trips may be designated by more than one organizational unit or depot. The service type is designated by the schedule origin, and as such a route may include trip sets that are assigned different day types for the same date. When a single dayType distinguishes the service provision for all the routes, then the routeDepotVersion is optional.
M	revisionNo	revisionNumber_id	This element is part of the identifying key of the schedule calendar date element. If multiple service revision documents are available, a rev-temporary or rev-permanent (i.e., scheduleVersionType) may supplant a scheduleVersionType equal to “original” or “suspended.”
M	scheduleVersionID	scheduleVersionID_id	This element is part of the identifying key of the schedule calendar date element. When a single dayType distinguishes the service provision for all the routes in a schedule version, then the identifying scheduleVersionID shall be used.
O	agencyID	agencyID_id	When more than one agency submits and stores a schedule calendar date data in the same SCD document, then the agencyID is needed. Otherwise it may be populated by the regional repository.

Usage and Examples of Schedule Calendar Date

This section describes several examples and issues that impact usage of the Schedule Calendar Date. The topics include:

- Service Precedence and Management of Schedule Version Components
- Schedule Note Versus Day Type Description
- Schedule Day
- Business Rules and Precedence Related to Assigning Day Types to Trips
- "Special" Day Type Descriptions and Extra Trips
- Physical Database Implementation

Service Precedence and Management of Schedule Version Components

The schedule version components consist of the Schedule Version, Revision and Route Depot Version. The schedule version fields at stored as an attribute group in the SDP XML Document header. The Schedule Revision is contained in the Agency Registration Branch of the Document. It describes the history of the schedule version. Most medium to small agencies will submit one document that covers their entire service for their quarterly schedule. Other agencies, supporting different modes, separately governed operating units or depots, or generating a significant number of routes, will generate multiple documents to register their schedules. When the Schedule Calendar Date is being populated or revised based on schedule component updates, the rules in Table 9-3 should be followed. (Low and High refer to precedence level.)

Table 9-3: Service Precedence and Management of Schedule Version Components

Sched. Ver. Type\Element	Schedule Version	Schedule Revision	Route Depot Version
original	Low	Low	Low
rev-temporary	Must be associated with an existing Schedule Version to be applicable	High Overrides service with scheduleVersionTypes: • original • rev-permanent Activation/deactivation dates describe dates of precedence; reverts back to last permanent schedule (original or rev-permanent).	High
rev-permanent	Must be associated with an existing Schedule Version to be applicable	High Replaces service with scheduleVersionType: • original Activation/deactivation dates describe dates of precedence; deactivation date should correspond to deactivationDate of original schedule version.	High
suspended	(Must be associated with an existing Schedule Version to be applicable.)	Suspends original; must be followed by a new schedule.	Not applicable

Schedule Note Versus Day Type Description

As recommend in Chapter 5.2, a note provides additional information to the customer, however, it does not aid the processing of information. As such, a note is acceptable as long as it is not a substitute for using or describing a day type. If notes are used, care should be taken to maintain the consistency between the note and day type code. Additional consistency checks may be needed to ensure that message and meaning matches the note and day type enumeration.

Schedule Day

The time used to designate service is defined as a 36 hour clock. The data type is a signed integer representing seconds past midnight. Negative values are associated with the day prior to midnight of the calendar date; values past 86400 seconds (24 hours) are associated with the day following the calendar date. As such, a trip assigned a day type code even if it logically falls on a day prior to or following the calendar date belongs to the assigned calendar date. For example, a day type designating special service for New Year's Day may include extra trips from 6 p.m. on December 31 to 3 a.m. on January 2.

Business Rules and Precedence Related to Assigning Day Types to Trips

A set of business rules are included to describe how day types will be automatically allocated to a calendar. The importance of these precedence rules is to help generate a timetable that groups and presents the information to the user in a format that is consistent and mitigates ambiguity. The basic requirement that drives these business rules is that the reserved set of day types should not overlap by days of the week. As such, for each route, any combination of non-overlapping day types is valid:

- Scenario 1: Weekday, Sat, Sun, Hol
- Scenario 2: Weekday, Mon-Sat, Sun, Hol
- Scenario 3: Weekday, Wk-closed, Sat, Sun, Hol
- Scenario 4: Sat, Sun, Mon, Fri, Tue-Thu, Hol
- Scenario 4: Sat, Sun, Mon, Tue, Wed, Thu, Fri, Hol

In all cases, special day types assume the highest level of precedence. A special day type must be assigned to a specific calendar date to be legitimate.

Scenario #1

- The service assigned to Weekday, Sat, Sun will be assigned to their representative days of the week.
- The service assigned to a Holiday day type will override the default day type based on the date when it is celebrated by the submitting agency during that calendar year (within the applicable schedule version).

Scenario #2

- A route or group of routes may be assigned to only one set of exclusive days of the week:
  - Mon-Sat and Sun will be assigned to their representative days of the week.
  - Weekday, Sat and Sun will be assigned to their representative days of the week.
  - All trips associated with a single route and route direction should conform to either [Mon-Sat and Sun] or [Weekday, Sat and Sun].

- The service assigned to a Holiday day type will override the default day type based on the date when it is celebrated by the submitting agency during that calendar year (within the applicable schedule version).

Scenario #3
- The service assigned to Weekday, Sat, Sun will be assigned to their representative days of the week.
- The service assigned to a Wk-closed day type will override a Weekday day type only. The Wk-closed will be assigned a date that corresponds with a school schedule submitted by an operating agency. The school schedule will only constrain the submitting agency schedules.
- The service assigned to a Holiday day type will override the default day type based on the date when it is observed by the submitting agency during that calendar year (within the applicable schedule version).

Scenario #4
- A route or group of routes may be assigned to only one set of exclusive days of the week:
  - Sat, Sun, Mon, Fri, Tue-Thu,
  - Sat, Sun, Weekday
  - Sat, Sun, Mon, Tue, Wed, Thu, Fri
- The service assigned to a Holiday day type will override the default day type based on the date when it is observed by the submitting agency during that calendar year (within the applicable schedule version).

"Special" Day Type Descriptions and Extra Trips

There are holidays and events when an Agency will run its "normal" schedule but add extra trips or trains to support a special event. For example, the Long Island Rail Road runs different service to Shea Stadium when the Mets are playing a home game at 5 p.m., a different service when the Mets are playing at 7 p.m., and yet another when the game is on the weekend, in the spring, summer and fall. In total, there are more than 70 different unique trip sets that operate on those "special" days. In reality, the schedule for those days remains the same for most service. Only a few train times are changed or trains added. There may be a limit to the number of special day types a downstream application can handle. To this end, the SDP enables alternate approaches for managing periods of service changes, extra trips or special events.

Within the context of choosing among these alternatives, one should recall these SDP requirements and precedent logic:
- The SDP limits the smallest unit of schedule component to a route, although the Route Depot Version will allow a route's trip set constrained by the day type as the smallest schedule component.
- The Route Depot Version and Schedule Revision may be constrained by activationDate and deactivationDate.
- A user-defined Day Type element (included the AgencyRegistration branch) may be constrained by beginTime and endTime.
- There are rules for logical precedence for day type enumerated types.

- There are rules for logical precedence for schedule version types (as assigned in Schedule Revision); the rules will apply to the most detailed Schedule Version component (e.g., schedule revision or route depot version).

Several scenarios handle various situations. These are described below:

Scenario #1—Typical Approach: Assign Day Type Codes to All Affected Routes and Related Service Components (Trip)

The typical and most often used by downstream applications is to generate the affected routes with trips regenerated under a different day type code. For example, if a local transit agency (agencyID=440) ran three trips (for a route) into New York City on a daily basis (dayType=weekday), although, on Thanksgiving Day they ran an additional trip on that route into New York City. The agency may use an existing SDP enumerated type (dayType=thanksgiving) to uniquely reference all the trips that operate service on that dayType, and they could specify all the day types associated with that trip in the Trip element. Specifically, the SDP document when detailing the trips for that Route would include four (4) trips, three (3) with dayType=weekday and dayType=thanksgiving, and one (1) trip[10] with just dayType=thanksgiving. The Schedule Calendar Date will have a total of 260 dates assigned service: 259 weekdays assigned a dayType code of "weekday," and the fourth Thursday of November will be assigned a dayType code of "special" of "thanksgiving,"

Scenario #2—Designate Time Range of Day Type Validity

This is a more complex approach than the previous scenario, and for that reason, special consideration must be used prior to using it. In this approach a special day type may be described that is constrained by beginTime and endTime. The logic for day type precedence must be implemented in the downstream application in order for this approach to work.

The Day Type element which is contained in the Agency Registration Branch of the SDP XML Schema includes timeBegin and timeEnd. If the precedence logic for day type is included in the downstream application data processing module, two Schedule Calendar Date elements with the same date, but different dayTypes may be included in the SDP XML Document for a given set of schedule version components. The first element would include normal operations, the second element will include a user defined dayType (e.g., Mets play night game on October 10) with the timeBegin and timeEnd specified (e.g., beginTime=5:30 a.m.; endTime=12 p.m.). Care must be applied when using this approach because all the trips that run on that schedule day, initiated after the beginTime and before the endTime for the specific scheduleVersionID, revisionNo and if appropriate routeDepotVersion, will be overridden by the designated dayType.

Scenario #3—RouteDepotVersion for Specific Time Periods

The RouteDepotVersion embedded in the Schedule Revision may be used to designate temporary service or alternate day types with different activation/deactivation dates within a Schedule Version. Where extra service is provided for a limited number or routes during the period between Thanksgiving and New Years, New Jersey Transit may designate changes to

[10] Three of the four trips would be the same three trips in the weekday list.

their route schedules based on activation and deactivation dates. Similar to Scenario #1, the entire set of trips operating on the active dates should be included in the SDP XML Document.

Physical Database Implementation

The White Paper on Schedule Calendar Date [DataExample_Schedule_Calendar_Date_v1.0.doc] includes the SCD XML Schema as well as a description of a physical database. The White Paper may be found on the SDP web site (see Appendix C for Resources).

Chapter 10: Advanced Topics on Select Data Concepts

In This Chapter

- Understand the requirements related to
  - Route and Timetable Header
  - Transfers: Transfer Cluster and Event Connection
  - Block (from Service Branch)
  - Route Grouping
- Learn how to apply these data concepts in the SDP

Advanced Topics

Advanced Topics describes SDP Data Concepts that are optional and designed for specific downstream applications. Topics include the following data concepts:

- Route and Timetable Header
- Transfer Cluster and Event Connection
- Block (for buses)
- Route Grouping

Why Advanced Topics on Select Data Concepts

The topics in this chapter are designated as *advanced* because the data concepts support key downstream applications that are not central to the initial scope of the Schedule Data Profile. They include topics such as:

- Timetable Header—captures the heading row of a public timetable. This is typically a manually derived set of data.
- Transfer Cluster and Event Connection—the places and times when connections may be made. The places refer to the Transfer Cluster, and the times refer to the Event Connection. These concepts differ from geographically and temporally calculated trip connections in that they are ***recommended*** transfer locations and times.
- Block (for buses only)—the daily work of a transit bus in revenue service. This is just a different way of grouping trip information.
- Route Grouping—a special designation for a route or group of routes. Again, a different way of grouping trip and pattern information.

Although these concepts may provide significant benefit to key downstream applications, most of these data concepts have not yet been validated in a demonstration environment or using real-world examples.

The data concepts may be found in different branches of the SDP XML hierarchy. Figure 10-1 highlights the position of these concepts within the SDP Schema. Under the Agency Registration branch may be found both the Route Grouping and the Timetable Header as a child element of the Route element. The Service branch includes both the Block (and Block Time) and Event Connection elements. The Transit Gazetteer includes the Transfer Cluster (and ConnectionSeg) element.

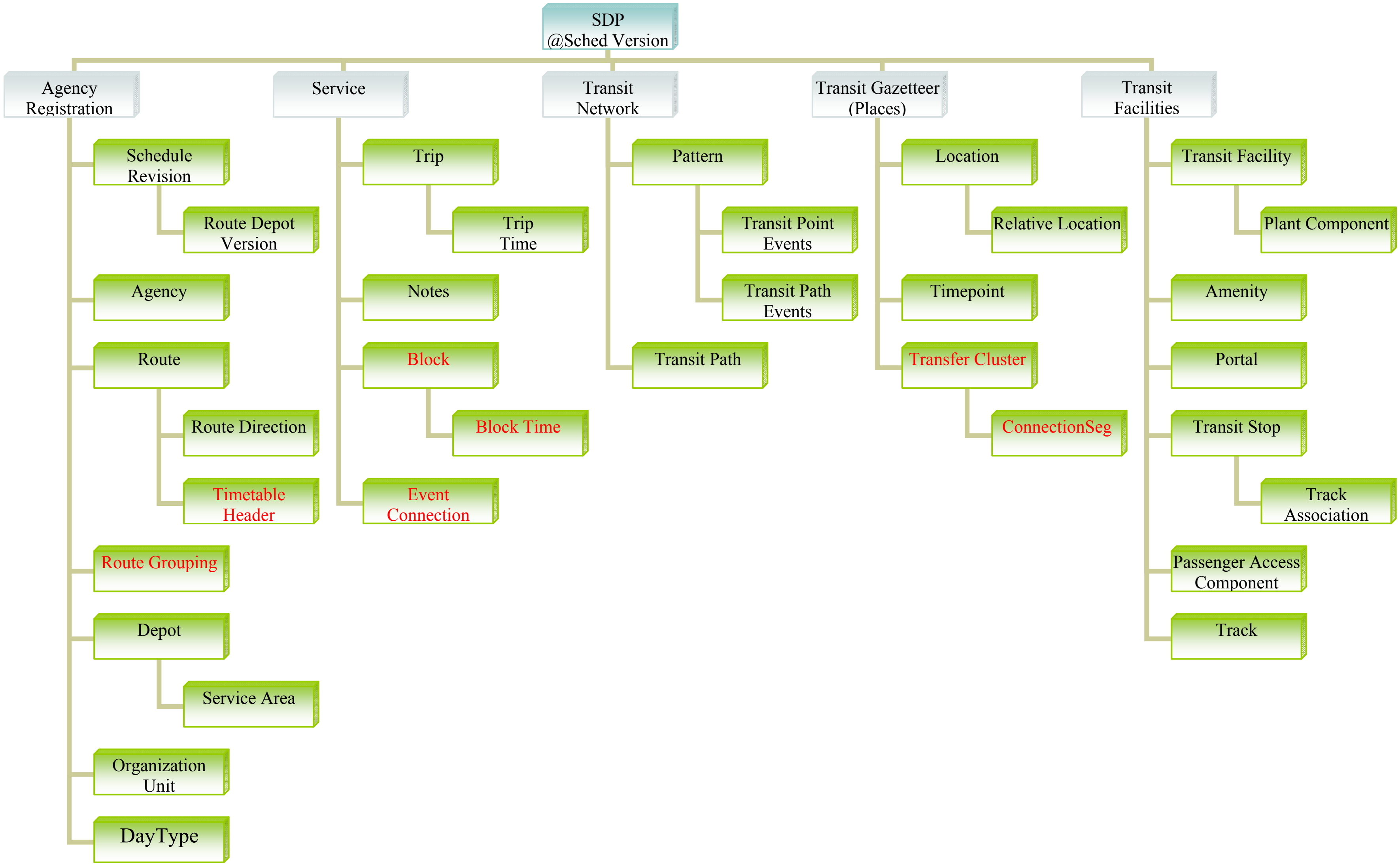


Figure 10-1: SDP XML Schema Organization with Advanced Topics highlighted in "red"

Section 10.1: Route and Timetable Header

In This Section

- Learn how to apply the Timetable Header Data Concept.
- Learn how to apply the Timetable Header element of the Route data concept.

Timetable Header Definition
The Timetable Header defines the header row for a table that contains public schedule information for a specific route. There should be one or two Timetable Header records (one for each direction) associated with each route.

What is the Timetable Header Data Concept?

A route timetable that is used by the public is an edited version of the full schedule. The public timetable summarizes each trip of all the patterns that are oriented in the same direction for a particular service type. An agency with three types of services—weekday, Saturday and Sunday—would generate three public timetables in each direction for each route. The Timetable Header data concept stores a header row of stop places for each route direction. The consolidation of the patterns of the same route direction is not necessarily a trivial exercise, neither is the reduction of timepoints or stops along each pattern. This exercise is typically a manual process performed by transit staff at each service change. The TimetableHeader data concept is a place to store summary header information with the schedule data so that when the pattern's events change, the header information may be updated simultaneously.

Timetable Header XML Schema Element Description

The Timetable Header concept is stored as a table in the CDRM. It is related directly to the Route it represents. This relationship is also passed on to the SDP XML Schema. The timetableHdr element is nested in the Route element (see Figure 10-2). Although optional in the Route structure, the timetableHdrList may contain many collections of headers. The rules related to applying the TimetableHeader Structure are as follows:

- Zero or more TimetableHeaderStructures may be included in the Route.
- The routeDirection should correspond to a routeDirection element in the routeDirectionList contained in the same Route.
- The timepointList is a locationID and its corresponding publicLocationDescription.

A number of assumptions may be inferred by downstream users or applications. For example:

- When present, it may be assumed that two timetableHdrList elements are included, one for each direction.
- When an agency prefers to display information about the stop, a stopID should be added to the timepointList.
- publicLocationDescription, although also contained in the Location table, is included in the TimetableHeader for the efficient processing of a downstream application that generates public timetables.
- This information may be derived (and stored at a later time) by an editing tool such as the Timetable Publisher application.

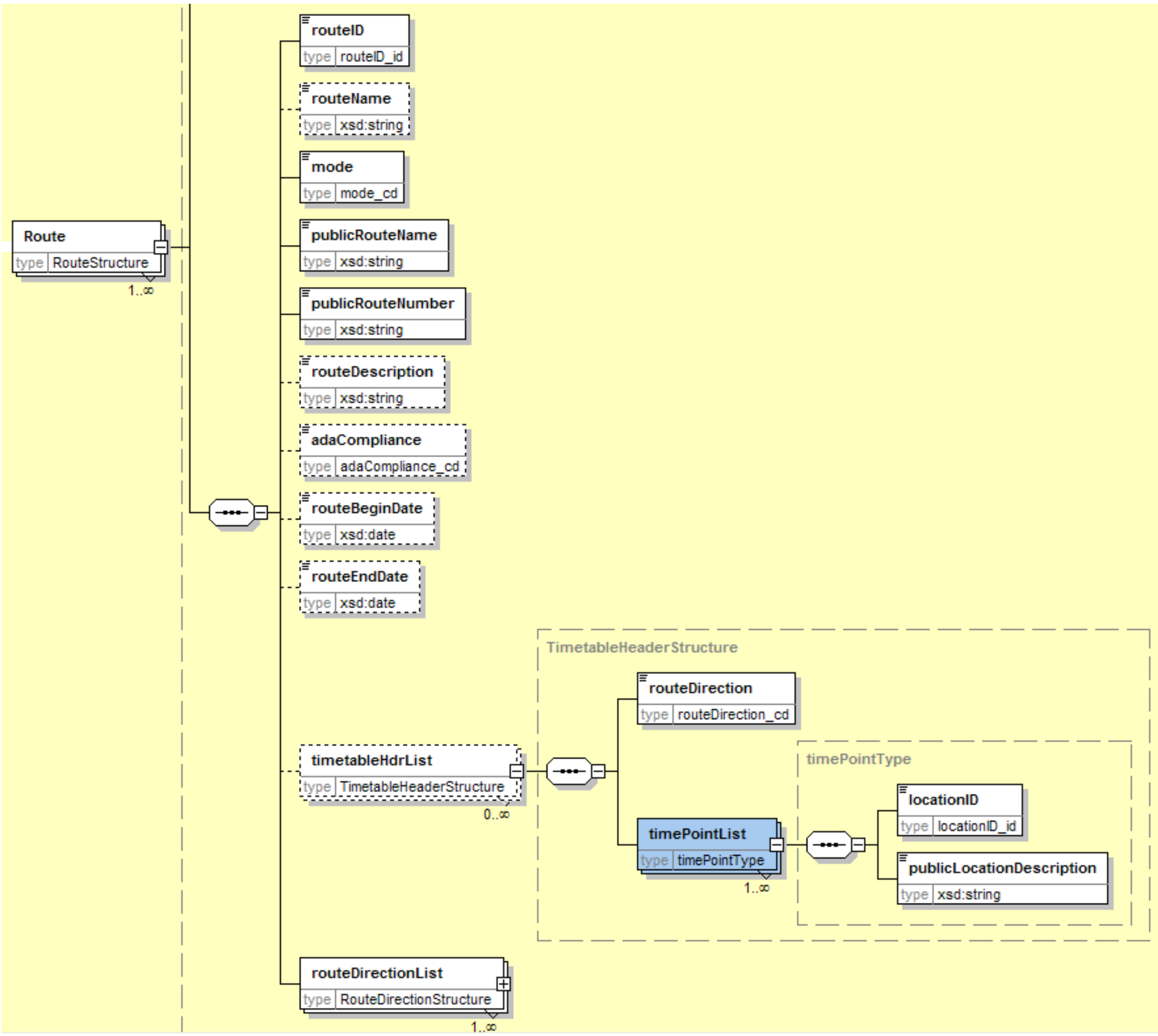


Figure 10-2: SDP XML Schema Fragment for Timetable Header

Detailed Data Descriptions and Guidance for the Timetable Header Element

This section provides data type information and guidance associated with TimetableHeader element described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

Table 10.1-1 incorporates TimetableHeader Guidance and Table 10.1-2 incorporates timePointType Guidance.

Table 10.1-1: Timetable Header Guidance

	Element Name	Type	Questions to Ask
M	routeDirection	routeDirection_cd	Use the routeDirection elements included in the routeDirectionList of the Route element. This element is used to select the trip/pattern that is oriented in the same direction.
M	timePointList	List of timePointType	As part of the data script, a timetable header may be generated. Generation requires review of each route's route direction, an aggregation of all events along all the patterns in the same direction, selection of the places along the pattern that are designated as timetable header transit places (locationID), and selecting the publicLocationDescription that matches the locationID (in the Location table).

Table 10.1-2 TimePointType Guidance

	Element Name	Type	Questions to Ask
M	locationID	locationID_id	A place that matches the Trip Time in order for a match with the Trip Time locationID to be made.
M	publicLocationDescription	string	A public name for the location that may be displayed on the timetable. This location description may be the same as a similarly named element in the Location table.

Examples of Timetable Header from a SDP XML Document

The Timetable Header documents the column headings for a route timetable. One Header record is associated with each direction. This example describes how the XML Schema will incorporate the Timetable Header record. The example shows how Suffolk County Transit (SCT) may define their Timetable Header for Route S29 Babylon serving the Walt Whitman Mall.

SCT may describe route-related information in the Route element and nest the TimetableHeader element under timetableHdrList (see SDP XML Schema description in Figure 10-2). The values for route are defined in Table 10.1-3.

Table 10.1-3: Example of Route Table

Route	Value
routeID	S29
routeName	S29 Babylon
publicRouteName	S29 Babylon—Walt Whitman Mall
publicRouteNumber	S29
routeDescription	S29 Babylon-Walt Whitman Mall
routeBeginDate	20041214 <!—yyyymmdd—>
routeEndDate	<!—null—>
timetableHdrList	<!—see below; only one direction is included in this example—>
routeDirection	[first, North], [second, South]

The Timetable Header derives its values from other elements in the schedule. For example, routeDirection should match the value in the Route/Route Direction, and the timepointList's locationID should match a locationID in a Route/Pattern/TransitPointEvent of a Pattern designated with the same route direction. For example, Route S29 may include five patterns, two patterns oriented in the first direction and three patterns in the second direction. Only the transitPointEvent elements that are associated with a Pattern specifying the second direction are considered for the header.

Given the requirements, SCT might specify a Timetable Header element that is described as an SDP XML Document fragment below.

```
<timetableHdrList>
    <routeDirection>second</routeDirection>
    <timePointList>
        <locationID>1204</locationID>
        <publicLocationDescription> Deer Park Rd and Old Country Rd </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1210</locationID>
        <publicLocationDescription> Deer Park Rd and Parsons Dr </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1217</locationID>
        <publicLocationDescription>2122 Deer Park Rd </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1223</locationID>
        <publicLocationDescription> Deer Park Ave and Weston Ave </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>156</locationID>
        <publicLocationDescription> Deer Park Ave and Strathmore Dr </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>984</locationID>
        <publicLocationDescription> Deer Park Ave and Montauk Hwy </publicLocationDescription>
```

</timePointList>
</timetableHdrList>

The Timetable Header for Route S29 in the second direction (i.e., South) for weekday schedules is illustrated in Figure 10-3. Note that the Dynamic Timetable Generator—DTG (the tool that generated the timetable illustrated in Figure 10-3) reads the header information, accesses the Trip information, and then matches the Trip's tripTime locationID to the header's locationID. A location that is included in the Trip but does not match the timetable header will be passed over. In addition, the application separated the service information by Route Direction (i.e., "Change Direction") and Day Type (i.e., "Change Day of Week").

Route Schedules — Suffolk County Transit

w to Read This Schedule | Complete Map | Routes | Fares | Rules and Tips | Contact Us

Route: S29 Route Map — South [Change Direction] — Weekday [Change Day of Week]

DEER PARK RD & OLD COUNTRY RD	DEER PARK RD & PARSONS DR	2122 DEER PARK RD	DEER PARK AVE & WESTON AVE	DEER PARK AVE & STRATHMORE DR	DEER PARK AVE & MONTAUK HWY
...	...	...	6:00	6:05	6:10
...	...	6:30	6:35	6:40	6:50
...	...	7:30	7:35	7:40	7:45
8:15	8:20	8:30	8:35	8:40	8:45
9:15	9:20	9:30	9:35	9:40	9:45
10:15	10:20	10:30	10:35	10:40	10:45
11:15	11:20	11:30	11:35	11:40	11:45
12:15	**12:20**	**12:30**	**12:35**	**12:40**	**12:45**
1:15	**1:20**	**1:30**	**1:35**	**1:40**	**1:45**
2:15	**2:20**	**2:30**	**2:35**	**2:40**	**2:45**
3:15	**3:20**	**3:30**	**3:35**	**3:40**	**3:45**
4:15	**4:20**	**4:30**	**4:35**	**4:40**	**4:45**
5:15	**5:20**	**5:30**	**5:35**	**5:40**	**5:50**
6:25	**6:30**	**6:35**	**6:40**	**6:45**	**6:50**
7:30	**7:35**	**7:40**	**7:45**	**7:50**	**7:55**

How to Read This Timetable | Complete Map | Routes | Fares | Rules and Tips | Contact Us

Figure 10-3: Example of Dynamic Timetable Generator Timetable for Suffolk County Transit

Section 10.2: Transfer Cluster and Event Connection Data Concepts

In This Section

- Learn about Transfer data concept requirements and issues. Learn about how the SDP supports Transfers. The key data concepts include
  - Transfer Cluster
  - Connection Segment, and
  - Event Connection
- Learn how to apply Transfer Cluster, Connection Segment and Event Connection to describe a Transfer.

Transfer Cluster Definition
A transfer cluster is a collection of transit stops where transfer between stops is convenient and scheduled.

Connection Segment Definition
A Connection Segment is a linear path allowing transit riders to move from one Transit Stop to another. The segment may be defined as a walking path, bike path, escalator or other modal connection. Attributes include distance, fromStop, toStop and connection instructions. Accessibility information in the form of obstacleTypes may optionally be provided for Connection Segments.

Event Connection Definition
An Event Connection is the provision for a connection between two route/trips at a trip time event. Connection types include [protected, recommended, scheduled].

Issues Related to Transfers

Transfer opportunities are easily derived by downstream applications by finding stops that are in close proximity and times where trips appear to allow for convenient connections. The requirements manifested in the SDP provide information on scheduled and recommended transfer locations and opportunities. A derived trip plan may not always generate a transfer that is convenient or viable, particularly if the path between trips is enabled by temporal events (e.g., gate open during business hours only, use of train in middle track as "platform bridge"). As such, the SDP data concepts allow for an agency to set recommendations for preferred transfer locations and opportunities. The SDP transfer provision includes a data concept in which a walking path (see Connection Segment Definition) may be defined that specifies access preferences.

Requirements for Transfer Cluster Data Concept

The requirements that drive the Transfer Cluster Data Concept are described in Table 10.2-1

Table 10.2-1 Transfer Cluster Data Concept Requirements

#	Category	Requirements
1	Unique identification, naming conventions, and references	• Transfer Clusters may contain both internal and public names and numbers. They are typically known through their Public Location Description (as described as publicLocationDescription in the Location Table).
2	Geometry and spatial characteristics	• A Transfer Cluster may share a location with one of the stops which is "clustered." • The location usually refers to a "centroid" or generalized location reference.

Table 10.2-1 Transfer Cluster Data Concept Requirements

#	Category	Requirements
3	Policy driven transfers	• A Transfer Cluster is defined for a set of stops where one or more agencies may coordinate schedule transfers between trips. The transfer may be based on a set of policies such as safety of boarding areas, connectivity, timed transfers, etc. • As a policy driven transfer, there may be special directions described for traversing from one boarding area to another. Several Connection Segments may be described between each stop pair. The connection may be accessible via elevator, ramp, etc. These are defined as a directed connection (from stop and to stop).
4	Associations	• A Transfer Cluster is associated with more than one Transit Stops (boarding area). • A Transfer Cluster may be indicated by the Trip Time timeEventType as a coordinated transfer location.

XML Schema Descriptions for TransferCluster and ConnectionSeg

The Transfer Cluster data concept includes a child element that describes one or more segments that connect the transfer locations (i.e., ConnectionSeg). Figure 10-4 depicts the SDP XML Schema fragment for TransferCluster and connectionList (instance of a ConnectionSeg).

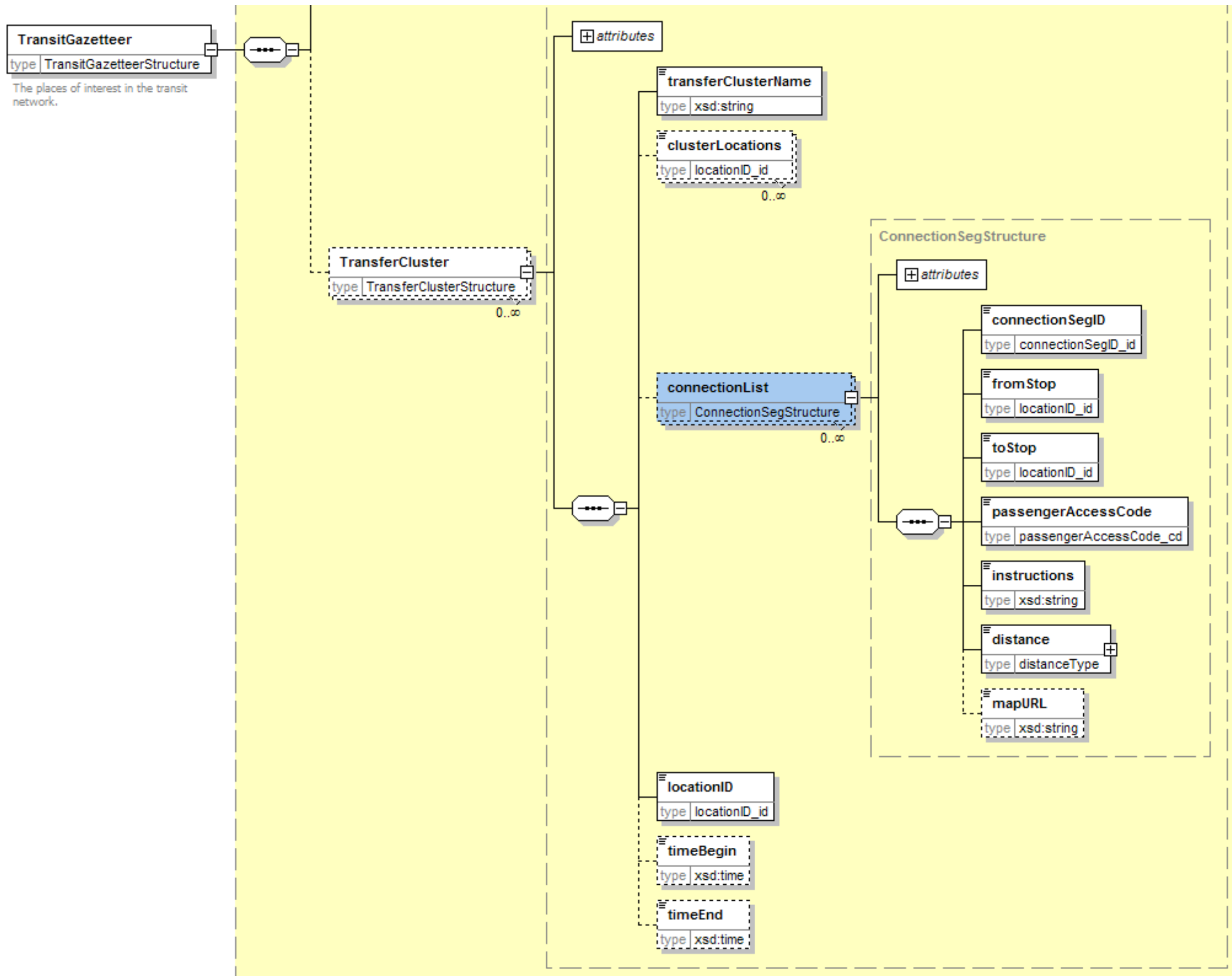


Figure 10-4: TransferCluster and ConnectionSeg Elements from SDP XML Schema

Specifically, the following rules were applied to build the XML Schema:

- The transferClusterName should be unique, and consistently spelled and capitalized. It is used by the EventConnection to reference the TransferCluster record.
- Transfer stops (referenced by locationID) are included by a series of clusterLocations.
- locationID describes the TransferCluster "centroid"; alternatively, the locationID may be a reference to one of the stops or a facility location.
- The TransferCluster may be temporal, and as such it is initiated and closed at specified begin (timeBegin) and end (timeEnd) times.
- The connectionList is a set of Connection Segments. Each Connection Segment is unique in each Transfer Cluster, as such, it may include an unique identifier (connectionSegID).
- The Connection Segment is a path between two Transit Stops (fromStop to toStop).
- The path may be described as an accessible or non-accessible path (passengerAccessCode)
- The instructions element contains human readable directions for the path from its origin to destination.
- The distance units may be specified as meters or feet.
- The mapURL is a link to a map or graphic of the connection path.

Conceptual Data Reference Model of the Event_Connection

The Event_Connection entity links two trips at a scheduled trip time pair. The transfer opportunity may be scheduled, coordinated or guaranteed (i.e., the connectionType code values). They may be from the same route, different routes or even different agencies.[11] The Event_Connection references the Transfer Cluster by its name (transferClusterName). The relationship among the Event_Connection entities is illustrated in Figure 10-5.

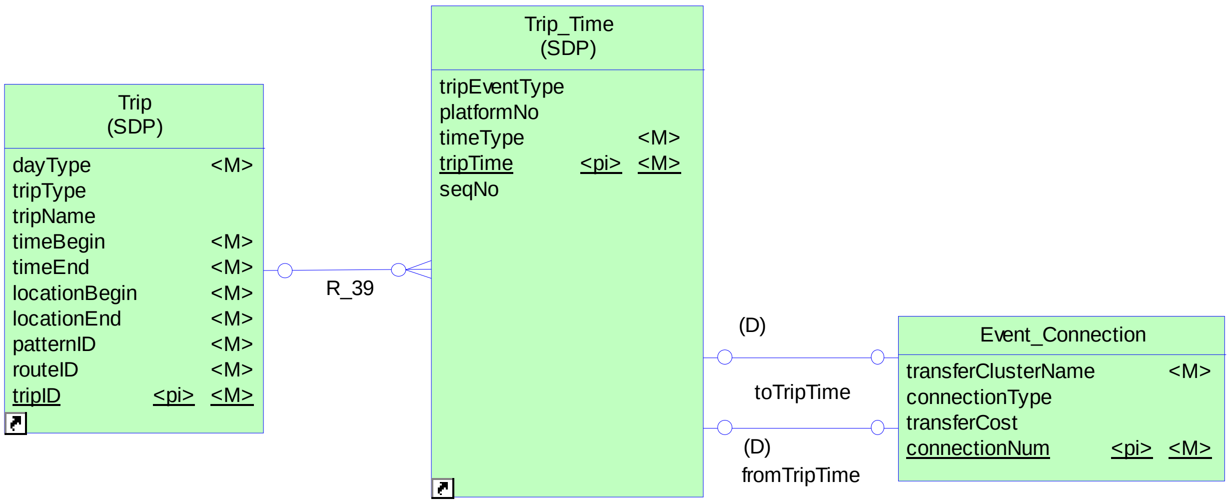


Figure 10-5: Event_Connection Conceptual Data Reference Model

Specifically, the Event_Connection CDRM represents:

A coordinated transfer (Event_Connection) may be designated between a pair of trips at a trip time from each trip. The time to transfer (e.g., 2 minute, 5 minute) is assumed from the fromTripTime to the toTripTime tripTimes. The location of each point and the path connecting them are described by the Transfer Cluster and referenced by the transferClusterName. The transferCost value is a description of the value or fare media that is required for this transfer. The connectionType values indicate if the trip is scheduled, coordinated or guaranteed.

The Event Connection table may include (from and to) agencyID when a multi-agency transfer is described.

XML Schema Description for EventConnection

The CDRM associates the Event_Connection with the Trip_Time. Because best practices in developing an XML Schema recommends that the schema limit the nesting elements (and flatten the hierarchy), the Event Connection is described as a child directly from the Service Branch. The rules applied to implement the EventConnection in the SDP XML Schema from the CDRM are as follows:

- Although the CDRM did not include foreign keys, the schema fragment does through the use of the XML Schema constraint -- KEYREF. The EventConnection must reference the Trip and Route from which the selected tripTimes derive. These are designated as “from” (fromRouteID, fromTripID) and “to” (toRouteID, toTripID).
- tripName (from and to) are included for readability and for use by rail transit to store the primary train number.

---

[11] A logical model that supports interagency connections should provide additional identifying keys into the Event_Connection entity to support connections between different agency trips.

- agencyID (from and to) should be included if the transfer occurs between two agencies. However, the SDP XML Document assumes the route XML Document contains only a single Agency's schedule.
- connectionType is an enumerated type.
- If included, transferClusterName must match the precise spelling and capitalization of the TransferCluster element's transferClusterName.

The XML Schema fragment for EventConnection is depicted in Figure 10-6.

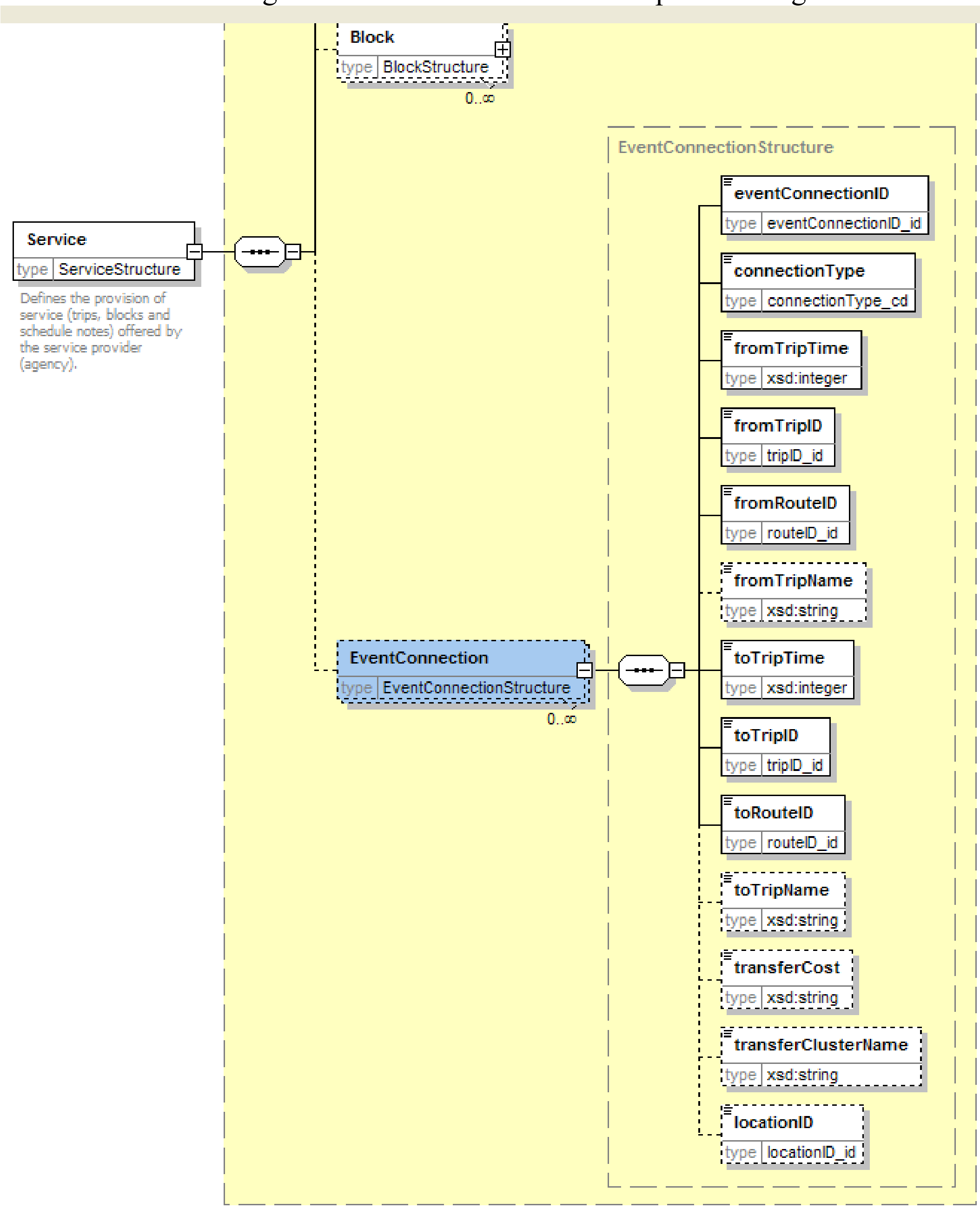


Figure 10-6 EventConnection SDP XML Schema Fragment

The schema fragment that depicts a guaranteed connection between two Metro North Railroad trains—Train 1530 (tripID 144711) on the New Heaven line and Train 1730 (tripID 144784) on the New Canaan line (at Stamford Station) is listed below. The transfer time between trips is 9 minutes (50220 sec – 49860 sec).

```
<EventConnection>
    <eventConnectionID>100</eventConnectionID>
    <connectionType>guaranteed</connectionType>
    <fromTripTime>49860</fromTripTime>
    <fromTripID>144711</fromTripID>
    <fromRouteID>3</fromRouteID>
    <toTripTime>50220</toTripTime>
    <toTripID>144784</toTripID>
    <toRouteID>4</toRouteID>
    <transferCost>0</transferCost>
    <locationID>124</locationID>
</EventConnection>
```

Detailed Data Descriptions and Guidance for the TransferCluster, ConnectionSeg and EventConnection

This section describes the format and guidance associated with TransferCluster, ConnectionSeg and EventConnection elements in the data concepts described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

The following tables provide guidance on TransferCluster (see Table 10.2-2), Connection Segment (Table 10.2-3), and EventConnection (Table 10.2-4).

Table 10.2-2: Transfer Cluster Guidance

Required	Element Name	Type	Questions to Ask
Transfer Cluster			
M	transferClusterName	string	A unique name by which the transfer cluster may be known. For example LIRR Jamaica Transfer
O	clusterLocations	list of locationID_id	These include the list of locations that are connected by this cluster. This may include multiple stops, facilities and

Table 10.2-2: Transfer Cluster Guidance

Required	Element Name	Type	Questions to Ask
			landmarks.
O	connectionList	ConnectionSeg	Descriptions of the paths between each transfer pair may be described using the connection segment.
M	locationID	locationID_id	This is a location identifier that designates the location of the transfer cluster. It may be the location of a facility like LIRR Jamaica Station or a shared facility like Mineola Intermodal Terminal.
O	timeBegin	time	Time transfer location is in service or accessible
O	timeEnd	time	Time transfer location is not in-service or is inaccessible.
O	@effectiveDate	date	[attribute] The date the record was inserted.
O	@endDate	date	[attribute] The date the record expires or becomes obsolete.

Table 10.2-3: Connection Segment Guidance

Required	Element Name	Type	Questions to Ask
Connection Segment			
M	connectionSegID	connectionSegID_id unique	An identifier that is stored by the SDP/TSDEA.
M	fromStop	locationID_id	The originating location.
M	toStop	locationID_id	The destination location.
M	passengerAccessCode	passengerAccessCode_cd	The type of access along the path.
M	instructions	string	A description of the directions along the connection path.
M	distance	distanceType	The distance between the origin and destination. An attribute, "units", may be designated for either feet or meters. The default units is ‘feet’.
O	mapURL	string	A link to a map showing the origin/destination; if this is generated, then it could be a web service that links the information from this element.
O	@effectiveDate	date	[attribute] The date the record was inserted.
O	@endDate	date	[attribute] The date the record expires or becomes obsolete.

Table 10.2-4: EventConnection Guidance

Required	Element Name	Type	Questions to Ask
Event Connection			
M	eventConnectionID	eventConnectionID_id	Insert a unique number or index in this field. Recommend a unique, sequential number be inserted for every event connection in the system throughout its lifecycle.
M	connectionType	connectionType_cd	This is a classification for connection type. These include: [scheduled, guaranteed, recommended].
M	fromTripTime	integer	The time (in schedule time) of the first trip's trip time (arrival or departure) to the stop. This time should match the originating trip's Trip.TripTime.tripTime value.
M	fromTripID	tripID_id	The tripID of the originating trip.
M	fromRouteID	routeID_id	The routeID associated with the originating trip.
O	fromTripName	string	A name used by the originating trip.
M	toTripTime	integer	The time (in schedule time) of the connecting trip's trip time (arrival or departure) to the stop. This time should match the connecting trip's Trip.tripTimeList.tripTime value.
M	toTripID	tripID_id	The tripID of the connecting trip.
M	toRouteID	routeID_id	The routeID associated with the connecting trip.
O	toTripName	string	A name used by the connecting trip.
O	transferCost	string	A description of the transit fare cost or media needed to transfer from the first to second trip.
O	transferClusterName	string	The name of the transfer cluster as defined in TransferCluster element (see TransitGazetteer branch).
O	locationID	locationID_id	The location of the connection. This may be either the originating trip or the connecting trip. When convenient, the Transfer Cluster or Transit Facility locationID should be used.

Examples of TransferCluster and ConnectionSegment

Because the Transfer Cluster encapsulates recommended transfer points and the paths between them, the Transfer Cluster may describe specific paths associated between two transit stops in a Transit Facility. In the example below, the TransferCluster element shows the TransferCluster between the Long Island Railroad platforms and Long Island Bus Terminal in the Mineola Intermodal Center (MIC).

More fully described in Chapter 9 on Transit Facilities, the MIC contains two facilities and three passenger access components (see Tables 10.2-5 for TransitFacility and Table 10.2-6 for selected PlantComponent element). There are two key connection segments for transfer from the LIB terminal to the LIRR platforms, these include the transfer to the eastbound platform and transfer to the westbound platform.

Mineola Intermodal Center

The transfer cluster composed of the Mineola Intermodal Center is described in Table 10.2-5 and its' Plant Components are described in Table 10.2-6.

Table 10.2-5: Mineola Intermodal Center TransitFacility Element with Two Nested TransitFacility Elements

Element name/values	LIB	LIRR	MIC
effectiveDate	2006-10-16	2006-10-16	2006-10-16
endDate	9999-12-31	9999-12-31	9999-12-31
transitFacilityID	MNLA	Mineola	MIC
locationID	MNLA	Mineola	MIC <master location>
facilityName	Mineola Bus Terminal	Mineola Station	Mineola Intermodal Center
facilityDescription	Mineola Bus Terminal	LIRR station serving Port Jefferson Branch	The Mineola Intermodal Center is on the south side of the LIRR track in the vicinity of Third Street between Third and Fourth Avenues and north of Old Country Road in Mineola.
owner	MTA	MTA	MTA
isPartOf	true	true	False
partOf	MIC	MIC	

Table 10.2-6: Selected List of Plant Component List in MIC TransitFacility

Element name/Values				
plantCompID	*componentID*	*plantCompType*	*plantCompDescription*	*locationDescription*
4	passAccessID >2	PassengerAccessComponent	North South Bridge with elevator linking Platform A, and Platform B, Bus Terminal and parking garage	N-S Bridge at west side of facility; entrance in Parking Garage.
From LIRR Transit Facility Plant Components				
1	stopID > MineolaA	TransitStop	Platform A	Inbound
2	stopID> MineolaB	TransitStop	Platform B	Outbound

The TransferCluster SDP XML Document fragment below includes the clusterLocations between the two facilities; these elements use location identifiers (locationID). The connectionList describes two paths between two stops (stopID), the first between the bus terminal (stop MNLA1) and Platform A, and the second between the bus terminal and Platform B. *The intention of the "instruction" element is that the description will be shown to a customer*. The map may be a compressed graphic or a link to a path and graphic reference, as suggested by the example. [The information in the SDP XML Document fragment is based on a photographs of the facility.]

```
<TransferCluster effectiveDate="2007-10-16" endDate="9999-12-31">
        <transferClusterName>Mineola Intermodal Center</transferClusterName>
        <clusterLocations> MNLA</clusterLocations>
        <clusterLocations>Mineola</clusterLocations>
        <connectionList effectiveDate="2007-10-16" endDate="9999-12-31">
                <connectionSegID>1</connectionSegID>
                <fromStop>MNLA1</fromStop>
                <toStop> MineolaA </toStop>
                <passengerAccessCode>elevator</passengerAccessCode>
                <instructions>"Take elevator on west side relative to bus berth 1 to
North-South Bridge.  Cross bridge, and take elevator on Rail Station side."</instructions>
                <distance units="feet">100</distance>
                <mapURL>www.mineolagraphic.net/mic.jpg</mapURL>
        </connectionList>
        <connectionList effectiveDate="2007-10-16" endDate="9999-12031">
                <connectionSegID>2</connectionSegID>
                <fromStop>MNLA1</fromStop>
                <toStop>MineolaB</toStop>
                <passengerAccessCode>elevator</passengerAccessCode>
                <instructions> Take elevator on north side relative to berth 1 to
North-South Bridge.  Cross bridge, and take elevator on Rail Station side."</instructions>
                <distance units="feet">140</distance>
                <mapURL> www.mineolagraphic.net/mic.jpg</mapURL>
        </connectionList>
        <locationID>MIC</locationID>
</TransferCluster>
```

Example of EventConnection with TransferCluster

In this example, the EventConnection shows a coordinated connection between a LIRR train and LIB trip at the Mineola Intermodal Center. [The example is derived from the Roosevelt East scheduled connection table.]

	TripTime locationID	tripTime	tripID	routeID
From	Mineola	23340 [6:29 AM]	1608 [604]	Port Jefferson
To	MNLA	23460 [6:31 AM)	2963	3484 [N22]

```
<EventConnection>
        <eventConnectionID>1</eventConnectionID>
        <connectionType>coordinated</connectionType>
```

```
    <fromTripTime>23340</fromTripTime>
    <fromTripID>1608</fromTripID>
    <fromRouteID>Port Jefferson</fromRouteID>
    <toTripTime>23460</toTripTime>
    <toTripID>2963</toTripID>
    <toRouteID>3484</toRouteID>
    <transferCost>Pay both LIRR and LIB fares</transferCost>
    <transferClusterName>Mineola Intermodal Center</transferClusterName>
    <locationID>mic</locationID>
</EventConnection>
```

Section 10.3: Block Data Concept

In This Section

- Learn about the Block Data Concept.
- Learn how to apply the elements in the Block data concept.
- Learn about the need for and application of the Block concept.

Block Definition
The daily sequence of revenue and nonrevenue trips assigned to a transit vehicle in revenue service from pull-out to pull-in.

Block Trip Time Definition
The path on which a transit vehicle in revenue service travels during the course of a day. Block Event Times (blockTimes) are part of an ordered set of events from beginLocation to endLocation. Each blockTime is associated with a specific event along the block, and is associated with one trip. The Block Event Time may be any type of transit feature type: timepoint, transit stop or other event (e.g., fare set change, headsign change) that is referenced by a Location (locationID).

Issues Related to the Block Data Concept

Bus and rail use the term *block* to mean very different concepts. In rail, block refers to a section of track, while in bus it means the daily work of a transit vehicle in revenue service from garage pull-out to pull-in. Although rail has a similar concept assigning train "work", a train's journey is closely tied to coordinating switches (e.g., controls) and assigning tracks, and consequently is not related to the types of information included in the SDP. On the other hand, many bus transit applications use the block structure to organize and predict bus arrival time at a stop for bus riders.

Requirements for Block Data Concept

The requirements associated with the Block Data Concept are listed in Table 10.3-1.

Table 10.3-1: Block Requirements

#	Category	Requirements
1	Uniqueness and identity	• A block is referenced by a unique identifier. The block identifier may be a combination of other identifiers such as route/run. • A block is assigned to a specific day or day type during a valid schedule version.
2	Temporal and Spatial Representation	• A block is a directed journey traversed by a transit vehicle while in revenue service. The journey is described by an ordered set of waypoints and times from the time a vehicle leaves its vehicle base until it returns to the vehicle base. [Note: during the course of a journey, a vehicle may be in revenue service, stopped, or traveling between revenue trips.] • At a minimum, a block is composed of two (origin and destination) or more sequence of locations (corresponding to pattern descriptions) and associated passing times (corresponding to trip descriptions), and any additional locations that represent events along the journey. • The "way points" consist of a unique, ordered sequence of transit point events such as timepoints, transit stops and events, and additional geo-located points that may be necessary to sufficiently describe the physical path. • A block has an origin and destination that typically correspond to a Transit Facility such as a vehicle base, depot or garage. The first and

Table 10.3-1: Block Requirements

#	Category	Requirements
		last (origin and destination) points constitute the termini of the block. • A block is typically assigned to one or more service or day types.
3	Alternative Geo-spatial composition as a set of transit paths	• Alternatively, a Block may be represented by a topologically ordered set of transitPaths (time point intervals, route segments). • The transitPath must include both tripTimes and transitPointEvents. • Each transitPath should be unique (for that day or day type) and associated with a trip identifier. • Note: the SDP transformation process will require that the points and times describing the transit path be reassigned to an ordered set of waypoints and times.
4	Sequential Waypoints and Passing Times	• Duplicate, sequential waypoints are allowed if they represent different types of passing points. For example, consecutive trips will share the same destination and origin, however, the destination will be described as an arrival time, and the origin of the next trip will be described as a departure time. • Duplicate, sequential waypoints are mandatory to differentiate the completion and commencement of different trips.
5	PTV assignment	• Schedulers typically assign a block to a vehicle type based on several characteristics supported by its fleet, such as capacity, seating, wheelchair lifts. • The actual assignment of the PTV is usually not made until the operator is dispatched and pulls out of the depot or vehicle base.

Conceptual Data Reference Model Description of Block

These requirements may be implemented in a CDRM depicted in Figure 10-7. The data model description follows.

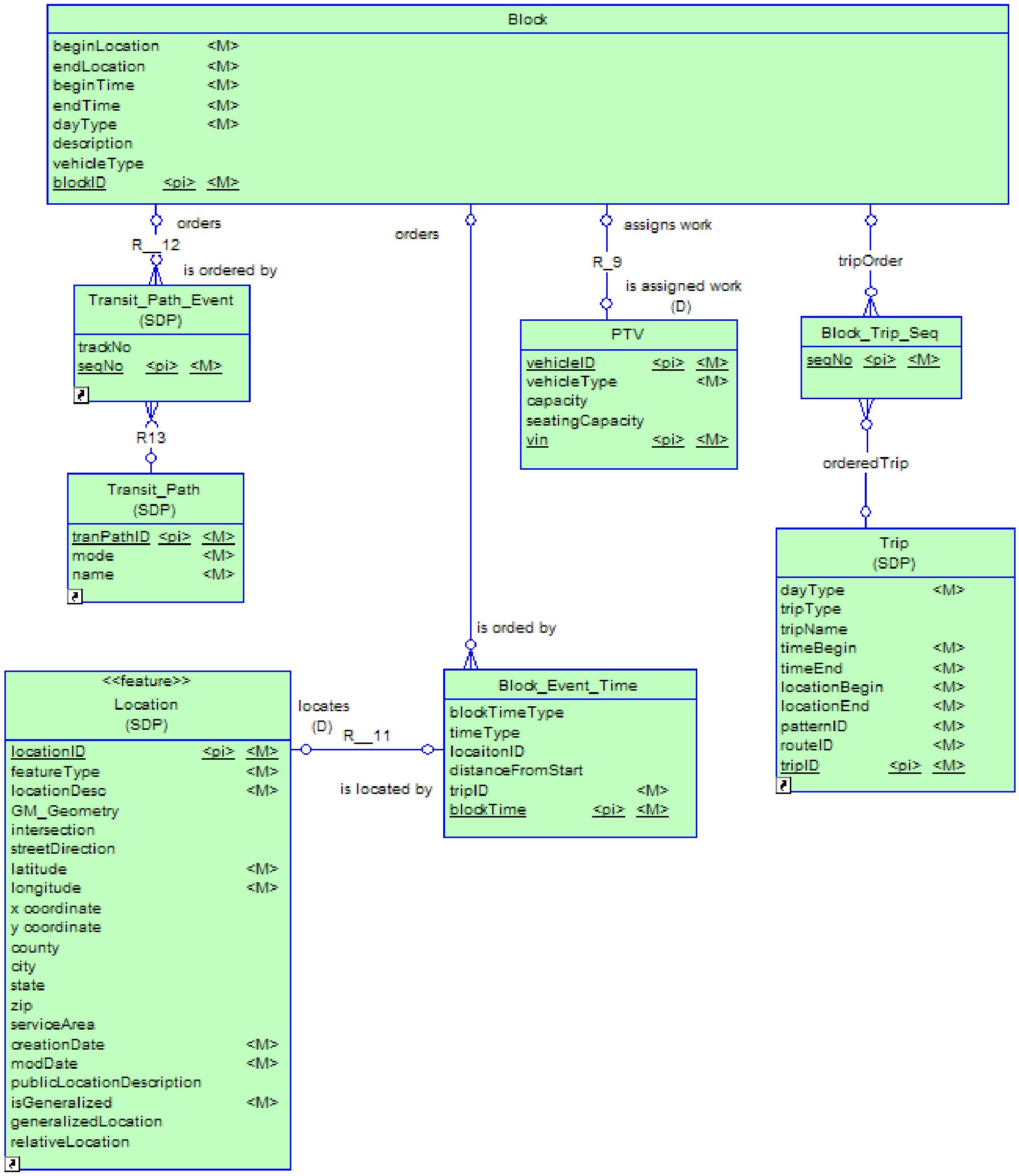


Figure 10-7: Block Data Model

A Block is described as the journey taken by a public transport vehicle in revenue service from when and where it leaves a transit facility until and where it returns to a transit facility. The actual PTV assignment may be made based on the required vehicleType. The path on which the vehicle travels is described by the BlockEventTimes, a series of coordinates, also linear references (distanceFromStart) from the beginLocation. The Block Event Times are ordered by blockTimes. Each blockTime is associated with a specific event along the block. Each Block Event Times is associated with one trip. The Block Event Time may be any type of transit feature type: timepoint, transit stop or other event (e.g., fare set change, headsign change) that is referenced by a Location.

Block Fragment of XML Schema Model

A number of rules and assumptions were used to implement the CDRM Block Data Concept as the Block and nested blockTimeList elements in the SDP XML Schema fragments (as shown in Figures 10-8 and 10-9). These include the following:

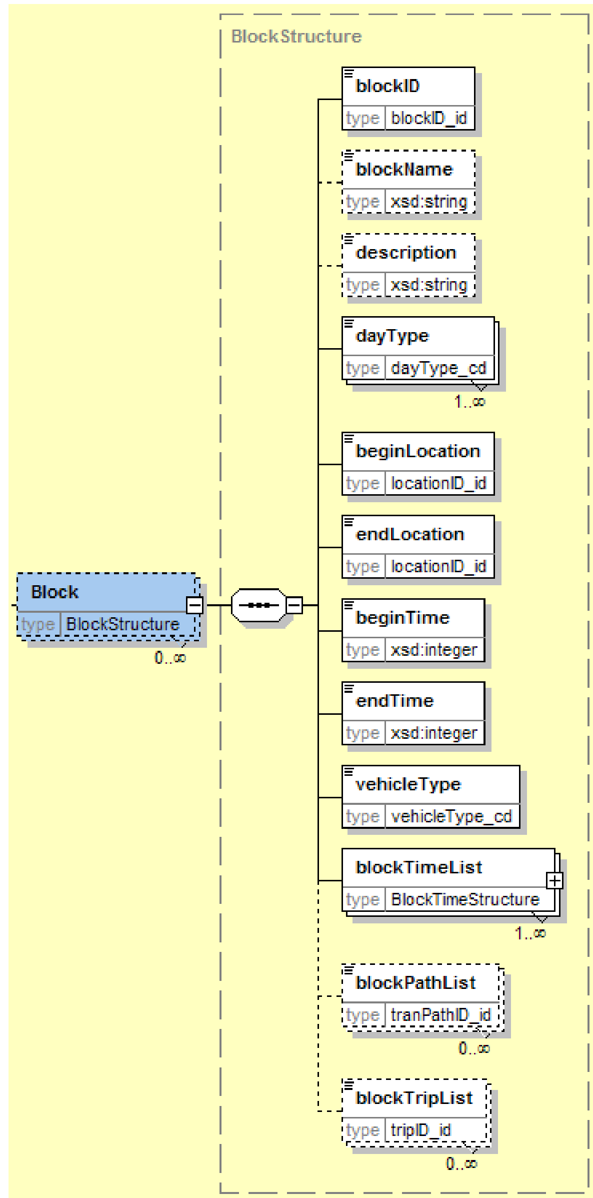


Figure 10-8: Block Element Fragment in SDP XML Schema

- blockID is unique across the data set;
- blockID, beginLocation, endLocation, beginTime, endTime, vehicleType and multiple blockTimeList nested elements must be included in Block structure.
- beginLocation and endLocation are coincident with the first and last blockTimeList (from blockTimeStructure) locationID element; beginTime and endTime are the same as the first and last blockTimeList blockTime element.
- A block is associated with the day type(s) on which it is assigned.

- blockPathList includes a sequential list of transitPathID.
- blockTripList includes a sequential list of tripID.

Note that the SDP does not include an actual transit vehicle assignment which is enabled by the CDRM Block data concept (see "PTV is assigned work to a Block"). The blockTimeList is equivalent to Block_Event_Time, the blockPathList is equivalent to Transit_Path_Seq, and the blockTripList is equivalent to Block_Trip_Seq.

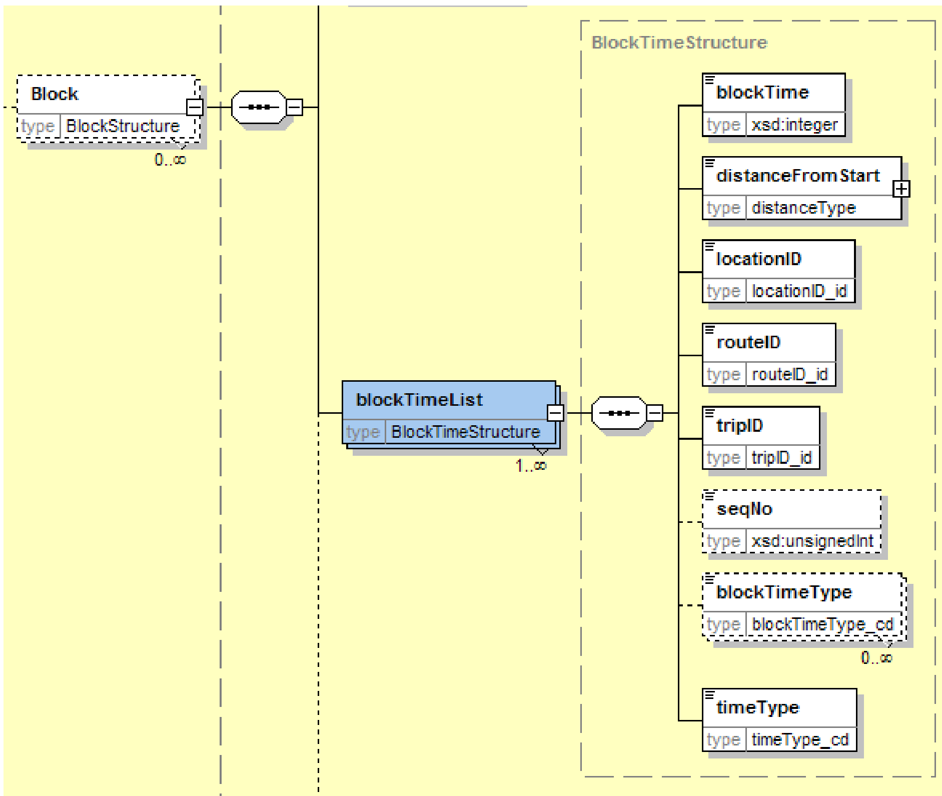


Figure 10-9: Nested blockTimeList Element Fragment in SDP XML Schema

Detailed Data Descriptions and Guidance for Block

This section describes the format and guidance associated with Block and BlockTimes elements in the data concepts described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

Table 10.3-2: Block Guidance

Required	Element Name	Type	Questions to Ask
Block (for bus only)			
M	blockID	blockID_id UNIQUE	Insert a unique number or character string for the blockID. Some agencies use a combination of the route and run numbers.
O	blockName	string	A block name is a character string that refers to the block. It is usually used for internal purposes.
O	description	string	A block description may be used to describe the vehicle journey and turn directions. It is typically used for internal purposes.
M	dayType	dayType_cd [1..∞]	The day type or day types to which it is assigned. Usually, the block assumes the day types of the service (trips) which compose its parts.
M	beginLocation	locationID_id	The beginLocation is typically the vehicle's pull-out location. It should correspond to the first timepoint of the first trip, (typically, the first trip is a non-revenue trip).
M	endLocation	locationID_id	The endLocation is typically the vehicle's pull-in location. It should correspond to the last timepoint of the last trip (typically, the last trip in a non-revenue trip).
M	beginTime	integer	The beginTime is typically the vehicle's pull-out time. It should correspond to the first timepoint of the first trip, (typically, the first trip is a non-revenue trip).
M	endTime	integer	The endTime is typically the vehicle's pull-in time. It should correspond to the last timepoint of the last trip (typically, the last trip in a non-revenue trip).
M	vehicleType	vehicleType_cd	The vehicleType is a general type of vehicle. This element supports internal fleet management strategies and in some cases customer-facing applications. The acceptable values mirror those in TCIP. They include additional values that may not be relevant for a block: ▪ twentyfiveRevenue ▪ thirtyRevenue ▪ fortyRevenue ▪ articulated ▪ cng ▪ lng ▪ supervisor

Table 10.3-2: Block Guidance

Required	Element Name	Type	Questions to Ask
			▪ police ▪ towTruck ▪ shelterService ▪ van ▪ passengerVehicle ▪ lightRail ▪ commuterRail ▪ heavyRail ▪ aircraft ▪ ferry ▪ transitPolice ▪ otherPolice
M	blockTimeList	List of BlockTimes	These are the Block Times along the block. They may be derived from the sequence of trips by ordering the tripTimes into the blockTimes.
O	blockPathList	List of tranPathID_id	The blockPathList describes an ordered list of the physical path traversed by a transit vehicle in a block.
O	blockTripList	List of tripID_id	The blockTripList describes list of the vehicle trips passed in the order they occur during the block.
Block Times (for buses only)			
M	blockTime	integer	The blockTime is the schedule time, that is, a signed integer of seconds past midnight of a schedule day.
M	distanceFromStart	distanceType: float	The value of (driven) distance from the beginning of the Block (blockLocation) to this Block Time location (locationID). The element includes an attribute that describes the unit of measure for the distance value. The allowed attributes include "feet" or "meters." The default enumeration type is “feet.”
M	locationID	locationID_id	The location identifier associated with the Block Time.
M	routeID	routeID_id	The route identifier tow which the trip is associated. The routeID provides an identifying reference for the tripID.
M	tripID	tripID_id	The trip identifier (tripID) to which the blockTime is typically associated. If the Block Time terminates one trip and begins another, then the timeType value should differentiate between an arrival/departure or begin/end Trip point.
O	seqNo	unsignedInt	The sequence number is a unique unsigned integer that is used to order the Block Time elements within a Block. The blockTime is ordered, although not sequentially numbered.

Table 10.3-2: Block Guidance

Required	Element Name	Type	Questions to Ask
			This value is available for use to associate the points of multiple temporal or linear paths.
O	blockTimeType	blockTimeType_cd	The blockTimeType designates key points that occur in a Block. These include: ▪ revenue ▪ pullIn ▪ pullOut ▪ beginInterDead ▪ endInterDead
O	timeType	timeType_cd	The timeType enumerated type uses the same values used in the Trip Time element.

Example of Block

The block is typically used for operational purposes, the blocking process assigns revenue rolling stock to planned service. In addition to estimating the number of revenue vehicles needed to deploy scheduled service, the block is also used to track the vehicle throughout the day. To this end, the block is used by many Automatic Vehicle Location (AVL) and real-time tracking systems to collect schedule adherence performance measures and to provide customers with estimated time of arrival predictions. The SDP block data concept is focused on supporting planned service data for these two functions.

Long Island Bus Block/Run 0101 for Weekday Service

This example shows the Block Structure for Block/Run 101 which covers Route 3210 (see Route example in Section 4.3).

Table 10.3-3: Trip Set for Run/Block ID for 101 From LI Bus Data

RouteID	TripID
3210	4575
3210	4576
3210	4577
3210	4578
3210	4579
3210	4580
3210	4581
3210	4582
3210	4583

```
<Block>
	<blockID>0101</blockID>
	<blockName>101</blockName>
	<dayType>weekday</dayType>
	<beginLocation>6</beginLocation>
	<endLocation>4918</endLocation>
	<beginTime>24180</beginTime>
```

```
        <endTime>51480</endTime>
        <blockTimeList>
                <blockTime>24180</blockTime>
                <distanceFromStart units="feet">0.00001</distanceFromStart>
                <locationID>6</locationID>
                <routeID>3210</routeID>
                <tripID>4575</tripID>
                <blockTimeType>pullOut</blockTimeType>
                <timeType>departure</timeType>
        </blockTimeList>
<!--more goes here -->
        <blockTimeList>
                <blockTime>51480</blockTime>
                <locationID>4918</locationID>
                <routeID>3210</routeID>
                <tripID>4583</tripID>
                <blockTimeType>pullIn</blockTimeType>
                <timeType>arrival</timeType>
        </blockTimeList>
        <blockTripList>4575</blockTripList>
        <blockTripList>4576</blockTripList>
        <blockTripList>4577</blockTripList>
        <blockTripList>4578</blockTripList>
        <blockTripList>4579</blockTripList>
        <blockTripList>4580</blockTripList>
        <blockTripList>4581</blockTripList>
        <blockTripList>4582</blockTripList>
        <blockTripList>4583</blockTripList>
</Block>
```

Section 10.4: Route Grouping Data Concept

In This Section

- Explore the Route Grouping Data Concepts.
- Learn when and how to apply the elements in the Organization Unit and Depot data concepts.

Route Grouping Definition
A realignment of a collection of patterns and/or trips known by a common name or number. The Route Grouping may be a "curtain" route, public timetable route, scheduler's route, or other collection of revenue service.

Purpose of Route Grouping in the SDP

Similar to a Route, a Route Grouping is a collection of patterns, trips or both. A route grouping is assembled for a specific purpose other then a typical Route definition.
For example, NYCT publishes for the public Routes S61 and S91, although their schedulers generate a Route 6191, and the bus carry a headsign displaying the "curtain" Route S61/S91. In this case, the routeID may be 6191, the public route number may be S61/S91. A Route Grouping record may aggregate all Route S61 patterns and trips and another may aggregate all S91 patterns and trips. Alternatively, the Route element may be indexed by 61 and 91, and the Route Grouping may aggregate the two routes into the S61/S91. Typically, the SDP recommends that the public-facing Route number is used in the Route element, and the ones used for internal and operational purposes use the RouteGrouping element designation.

Requirements for Route Grouping Data Concept

The requirements associated with the Route Grouping Data Concept are listed in Tables 10.4-1.

Table 10.4-1: Route Grouping Requirements

1	Definition	• A line or route grouping is a collection of patterns or trips that are grouped together for a particular purpose. - A set of patterns or trips may be grouped together operationally in order to closely coordinate their scheduled headways along a common alignment or carriageway. - Public timetables group routes together to communicate the frequency of service of routes that share a common corridor before they branch. o There are other types of groups and some of these are listed in Route Type
2	Uniqueness and identity	• A route grouping is referenced by a unique identifier which is distinct from a Route. Each route grouping identifier should be uniquely defined within the schedule version, revision and activation/deactivation dates. The route grouping identifier may be a combination of other identifiers such as two route identifiers or a partition of a single route identifier. • A route grouping may be identified by its duration during a schedule version such as its activation and deactivation dates.

Conceptual Data Reference Model Description for Route Grouping

The Route Grouping requirements from Table 10.4-1 may be implemented in the CDRM as depicted in Figure 10-10. The data model description (as excerpted from the *Functional Requirements* document) follows.

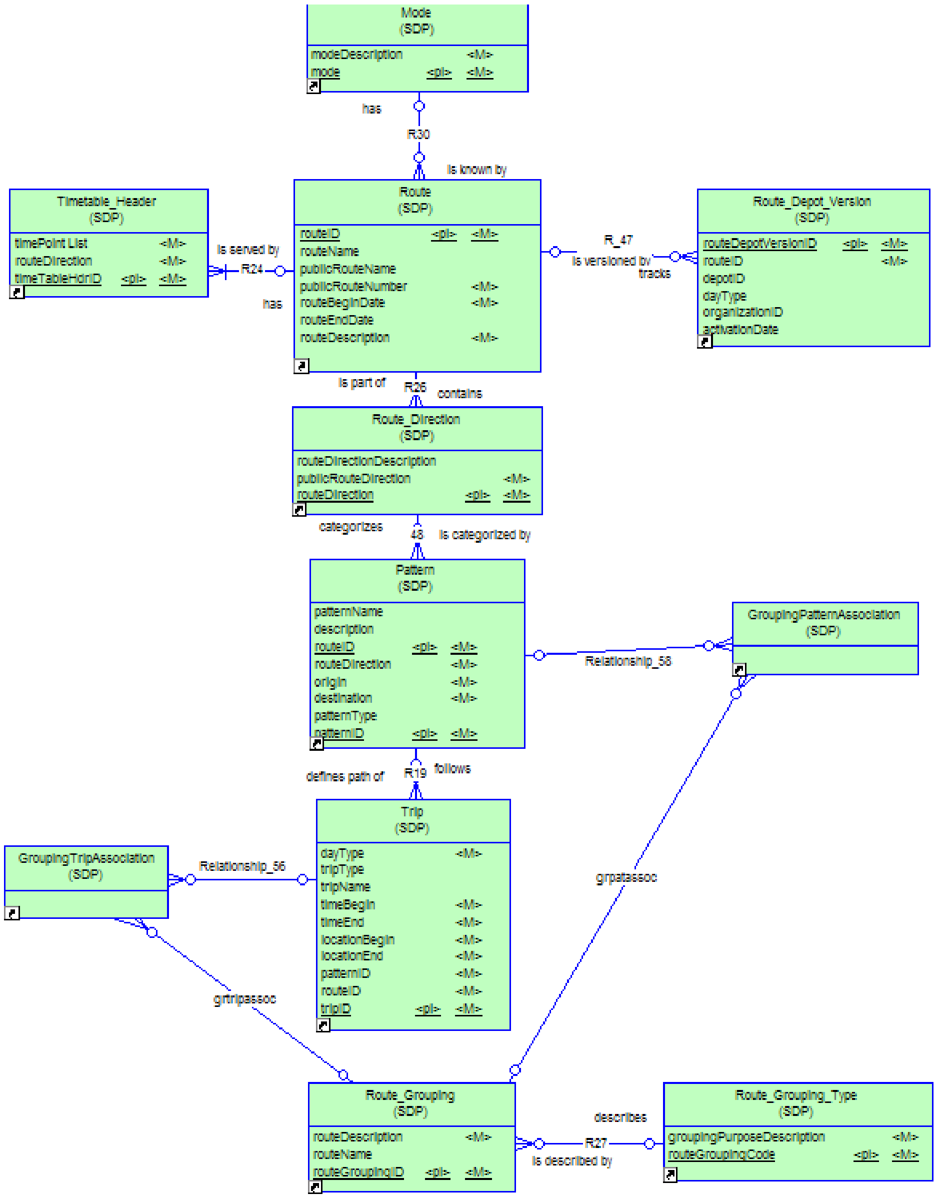


Figure 10-10: Route Grouping Data Model

"A Route is a collection of patterns classified by their route direction in scheduled service with a common identifier. As shown, Route is an entity associated with patterns and related trips. One or more directed patterns may optionally be associated with a Route, and one Route categorizes one or more patterns through a route direction. One or more trips may optionally be associated with each pattern...

"A Route is valid during its designated period (routeBeginDate to routeEndDate) that falls within a valid schedule version.

"Similar to a Route, a Route Grouping is a collection of patterns, trips or both. A route grouping is assembled for a specific purpose other then a typical Route definition."

Route Grouping Excerpt of XML Schema Model

The following rules and assumptions were used to implement the CDRM Route Grouping Data Concept as the RouteGrouping element in the SDP XML Schema:

- A unique identifier, routeGroupingID, is associated with the RouteGrouping element.
- The element includes XML attributes to designate the placement and deactivation dates of the element.
- The element is associated with a routeName and sometimes with a routeDescription (either for operational purposes or for the public).
- The routeGroupingCode describes the purpose for the route grouping. The code values include: curtain, public, line. Other codes may be designated by code values above an integer value of 5.
- The groupingSetList builds the associations between Route_Grouping and its related Patterns (GroupingPatternAssocation) and Trips (GropuingTripAssociation).
  - Through the groupingSetList, a list of applicable route-pattern (patternSet) and/or related route-trip identifiers (tripSet) are associated with each RouteGrouping element.

Figure 10-11 depicts the RouteGrouping element portion of the SDP XML Schema as it was derived from the CDRM.

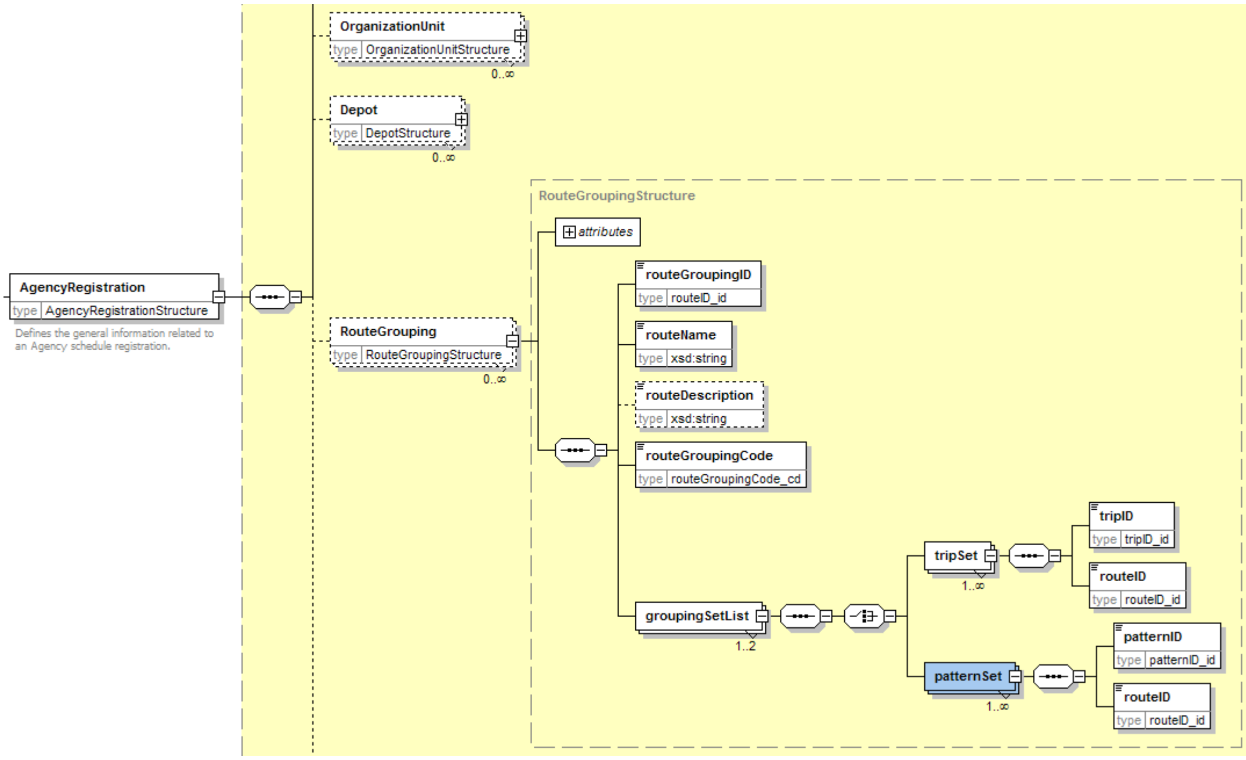


Figure 10-11: SDP XML Schema Fragment of Route Grouping

Detailed Data Formats and Guidance for the RouteGrouping Element

This section describes the description and guidance associated with RouteGrouping element in the data concepts described above. The guidance for each element is consolidated into Table 10.4-2 with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

Table 10.4-2: Route Grouping Guidance

Required	Element Name	Type	Questions to Ask
RouteGrouping structure			
M	routeGroupingID	routeID_id UNIQUE	The routeGroupingID is unique in the schedule version for both the Route Grouping and Route elements.
M	routeName	string	If the route grouping element is used for the public, then routeName may also be used for as a public name like the Route element – publicRouteName. Otherwise, it is used by schedulers or for other internal and operational purposes. The name should be formatted for the purpose of the route grouping.
O	routeDescription	string	Similar to the routeName, the routeDescription is typically used for internal and operational purposes. The description should be formatted for the purpose of the route grouping.
M	routeGroupingCode	routeGroupingCode_cd	The routeGroupingCode is used to describe the purpose of the route. Curtain and Public are public-facing route identifiers and Line is a scheduler's artifact.
M	groupingSetList	[1] tripSet; and [2] patternSet	Both elements (and no more then the two elements) should be included in the groupSetList.
tripSet element			
M	tripID	tripID_id	The tripID is associated with an existing route designated in the Trip element. The trips listed in this set should correspond to the patterns identified in the patternSet
M	routeID	routeID_id	The routeID is needed to distinguish the trip identified above.
patternSet element			
M	patternID	patternID_id	The patternID is associated with an existing route designated in the Pattern element.
M	routeID	routeID_id	The routeID is needed to distinguish the pattern identified above.

Examples Using Route Grouping

There are several types of route groupings. The major categories that will be described in this section are two scenarios that exist in the downstate NY region. The first scenario is when a single route is separated into two routes for a singular purpose like branding an express versus local service in the same scheduled route. The second alternative is when two routes are merged into a single route perhaps for operational purposes or because they share a common trunk. New York City Transit and New Jersey Transit have several examples that cover these scenarios.

Scenario #1: Merging two routes into a single route

New Jersey Transit Route 78 timetable carries two designated scheduled routes: #78 and #378. #78 is the Newark/Penn Station to Secaucus-Harmon Cove via Plaza Drive and Meadowland Parkway, and #378 goes directly from Newark/Penn Station to Secaucus-Harmon Cove (also called express service). In the scheduling system, the holiday service may be scheduled differently, for example, Route 378 is not operated on the Friday after Thanksgiving, MLK Day and President's Day. This Route Grouping scenario is described in Table 10.4-3 below.

Table 10.4-3: Route Grouping Example for NJ TRANSIT Route 78 Timetable

Route Grouping	Record 1
routeGroupingID	78
routeDescription	Serving Newark Secaucus Harmon Meadow Harmon Cove
RouteName	78 Newark Secaucus

Route Grouping Type	Type associated with routeGrouping ID = 78
routeGroupingCode	1
groupingPurposeDescription	Similar origin and destination (local and express service) subject to different holiday schedules; may be used to generate timetable.

Partial Set of Trips Associated with routeGroupingID=78								
tripID	tripType	dayType	timeBegin	timeEnd	locBegin	locEnd	routeID*	pattID*
7801	1	1	5:57	6:54	1001	2156	78	7811
7802	1	1	6:35	7:32	1001	2156	78	7811
7803	1	1	7:00	7:57	1001	2156	78	7811
7804	1	1	7:20	8:17	1001	2156	78	7811
7805	1	1	7:50	8:47	1001	2156	78	7811
7806	1	1	8:20	9:24	1001	2156	78	7811
7807	1	1	8:50	9:47	1001	2156	78	7811
7808	1	1	9:20	10:23	1001	2156	78	7811
7809	1	1	11:00	12:02	1001	2156	78	7811
7810	1	1	13:00	13:57	1001	2156	78	7811
7811	1	1	15:00	15:56	1001	2156	78	7812
37801	2	1	16:07	16:39	1001	2156	378	37811
37802	2	1	16:35	17:07	1001	2156	378	37811

*needed for physical database; patternIDs, locationBegin and locationEnd are arbitrary numbers.

Where tripType 1=local and tripType 2=express

Where dayType 1=weekday

Scenario #2: Composite Route Separated in Route Grouping

In the second scenario, NYCT may separate a single route into two route grouping for the purpose of branding local and limited stop services. In the example below (see Table 10.4-4) the composite route 6292 is divided into two routes in order to brand the limited stop route with its own public route number and name.

Table 10.4-4: Route Grouping Example for NYCT Composite Route S62/S92

Route Grouping	Record 1	Record 2
routeGroupingID	S62	S92
routeName	Local for Route S62/S92	Limited Stop Service for S62/S92
routeDescription	S62 Local Travis-St. George Ferry Terminal	S92 Limited Stop Travis St.-George Ferry Terminal

Route Grouping Type	routeGroupingCode	groupingPurposeDescription
Type associated with routeGrouping ID = S92	curtain	Express Service
Type associated with routeGrouping ID = S62	curtain	Local Service

Partial Set of Trips Associated with routeGroupingID=S62								
tripID	tripType	dayType	timeBegin	timeEnd	locBegin	locEnd	routeID*	pattID*
6201	1	1	00:20	12:51	2345	5657	6292	6202
6202	1	1	1:20	1:51	2345	5657	6292	6202
6203	1	1	2:20	2:51	2345	5657	6292	6202
6204	1	1	3:20	3:51	2345	5657	6292	6202
6215	1	1	3:50	4:21	2345	5657	6292	6202
6206	1	1	4:20	4:51	2345	5657	6292	6202
6207	1	1	4:50	5:21	2345	5657	6292	6202
6209	1	1	5:10	5:41	2345	5657	6292	6202
6208	1	1	5:25	5:56	2345	5657	6292	6202
6210	1	1	5:40	6:12	2345	5657	6292	6202
6211	1	1	5:55	6:29	2345	5657	6292	6202
6212	1	1	6:21	6:39	4556	5657	6292	6204
6213	1	1	6:41	6:59	4556	5657	6292	6204
6214	1	1	6:56	7:16	4556	5657	6292	6204
6205	1	1	7:11	7:32	4556	5657	6292	6204
6221	1	1	7:26	7:47	4556	5657	6292	6204
6216	1	1	7:41	8:02	4556	5657	6292	6204
6217	1	1	7:56	8:17	4556	5657	6292	6204
6218	1	1	8:11	8:32	4556	5657	6292	6204
6219	1	1	8:20	8:50	5766	5657	6292	6207
6220	1	1	8:21	9:00	2345	5657	6292	6202

Partial Set of Trips Associated with routeGroupingID=S92								
tripID	tripType	dayType	timeBegin	timeEnd	locBegin	locEnd	routeID*	pattID*
9201	2	1	6:10	6:47	2345	5657	6292	6203
9211	2	1	6:28	7:05	2345	5657	6292	6203
9202	2	1	6:43	7:21	2345	5657	6292	6203

9203	2	1	6:58	7:36	2345	5657	6292	6203
9204	2	1	7:12	7:51	2345	5657	6292	6203
9212	2	1	7:27	8:06	2345	5657	6292	6203
9205	2	1	7:42	8:21	2345	5657	6292	6203
9206	2	1	7:57	8:36	2345	5657	6292	6203

*needed for physical database;
patternIDs, locationBegin and locationEnd are arbitrary numbers.

Where tripType 1=local and tripType 2=express.
Where dayType 1=weekday

Example of the RouteGrouping XML Excerpt

This is a XML fragment from Scenario 2 Composite Route Separated in Route Grouping, Record 1 from Table 10.4-4.

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
	<routeGroupingID>S62</routeGroupingID>
	<routeName> Local for Route S62/S92</routeName>
	<routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
	<routeGroupingCode>curtain</routeGroupingCode>
	<groupingSetList>
		<tripSet>
			<tripID>9201</tripID>
			<routeID>6292</routeID>
		</tripSet>
	</groupingSetList>
	<groupingSetList>
		<patternSet>
			<patternID>6203</patternID>
			<routeID>6292</routeID>
		</patternSet>
	</groupingSetList>
</RouteGrouping>
```

The SDP document fragment may also be documented with either the tripSet or the patternSet, as follows:

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
	<routeGroupingID>S62</routeGroupingID>
	<routeName> Local for Route S62/S92</routeName>
	<routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
	<routeGroupingCode>curtain</routeGroupingCode>
	<groupingSetList>
		<tripSet>
			<tripID>9201</tripID>
			<routeID>6292</routeID>
		</tripSet>
```

```
        </groupingSetList>
</RouteGrouping>
```

Or

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
        <routeGroupingID>S62</routeGroupingID>
        <routeName> Local for Route S62/S92</routeName>
        <routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
        <routeGroupingCode>curtain</routeGroupingCode>
        <groupingSetList>
                <patternSet>
                        <patternID>6203</patternID>
                        <routeID>6292</routeID>
                </patternSet>
        </groupingSetList>
</RouteGrouping>
```

Chapter 11: SDP Document Conformance Requirements

In This Chapter

- Learn about the requirements and tests required to ensure SDP Document Conformance; these include validity checks, XML validation, and conformance with document naming procedures.
- Learn about how to create a SDP Conformance Profile for a downstream application.
- Understand the issues related to extending or constraining the SDP XML Schema.

Conformance Definition

The act of meeting requirements or bringing about accord or compliance.

This chapter includes sections on:

- Scope of SDP Conformance Requirements
- SDP XML Document Conformance Requirements
- Conformant Profile Development

Scope of the SDP Conformance Requirements

Conformance requirements are the set of tests that must be met in order for the SDP Document to be conformant to the SDP. These conformance requirements incorporate the SDP XML Schema Document's well-formedness and validation requirements, that is, the SDP Document must comply with the organization and format of the SDP XML Schema. However, there are other requirements related to the data which must also be met. These include testing the referential integrity and consistency of the native data inserted into the SDP, and confirming that the SDP Document was named using the prescribed naming procedures.

The original SDP XML Schema, based on the SDP's Conceptual Data Reference Model (CDRM), ensures that a baseline set of requirements will be met that support many downstream applications. However, new applications may be developed and previously existing applications may need additional information to function properly over time. For example, a category of data may be present in the original SDP XML Schema as an optional element, but now it needs to be mandatory. Alternatively, the information may be available, but not in a format that is usable by the downstream application. The approach used to develop the SDP XML Schema is flexible; it allows a version of the schema to be constrained or extended to meet the changing needs of downstream applications.

This chapter discusses two topics on:

- How conformance is met, that is, the tests that must be implemented to achieve conformance (section 11.1); and
- How to create a Conformance Profile that meets downstream application data requirements within the semantic requirements of the conceptual data model (section 11.2).

Detailed examples are included or referenced in both sections to help understand how to apply the requirements.

Section 11.1: SDP XML Document Conformance Requirements

In This Section

- Learn about generic XML Schema standard conformance for SDP XML Documents.
- Learn about data validity checking for the content of the SDP XML Document.
- Understand the principles of naming a SDP XML Document.

Conformance Requirements
SDP Conformance is the set of requirements that must be set by an SDP XML Document. These requirements include:

- Checking for data validity and integrity;
- Validation of the SDP XML document for well-formedness and validity (against a valid SDP Schema);
- Adoption of SDP Document naming procedures.

Formal Conformance Requirements are composed of four statements:

- Definition of Conformance
- Conformance Situation
- Conformance Criteria
- Test Methods

The set of requirements that constitute conformance are included in *Definition of Conformance*. The statement typically references the set of requirements in the standard or specification. In the case of SDP, the SDP XML Schema and base standards define the requirements.

The *Conformance Situation* describes the constraints or extensions related to the set of requirements. These rules define specific situations where elements may be inserted or removed, and still be conformant to the specification.

Conformance Criteria describes specific rules that should be tested as part of the conformance requirements, as such, each criteria is associated with a specific Conformance Requirement (as listed in the Definition of Conformance).

Finally, *Test Methods* describes how the Conformance Criteria shall be tested.

Definition of Conformance

Conformance to the SDP XML Schema is defined as adopting the following major conformance requirements (CR):

CR.1 The SDP XML Document shall be “well-formed” and “valid" as validated against an approved SDP XML Schema using two or more XML Schema Validation tools. “Well-formed” and “Valid” refer to the XML Schema 1.0 (or the latest version).

CR.2 The SDP XML Document shall be checked for valid referential integrity among unique keys and referenced keys (e.g., primary and foreign keys).

CR.3 The SDP XML Document shall conform to the SDP XML Document naming conventions.

Conformance Situation

A SDP XML Document may be extended or constrained and still be conformant if the SDP XML Schema does not abrogate or change any of the baseline (version 1.0 or latest version) mandatory schema requirements.

The following changes are **not** allowed:

- Change the order of parent or child elements
- Change a “mandatory” element to optional (i.e., change minOccurs value to “0”)
- Change a single occurrence of an element to multiple occurrences
- Remove or rename any of the elements in the baseline schema
- Change the type of each simpleType

The following changes are allowed:

- Restrict the size or range of an element type (e.g., from string to nine character string)
- Mask a string type
- Change an “optional” element to mandatory (i.e., change minOccurs=“0” to a number greater than zero)
- Extend the value list of an enumerated type (code)
- Add optional attributes to an element (i.e., use=“optional”)
- Under certain circumstances, include additional “optional” elements (with element definitions) at the end of a complex type description

Conformance Criteria

The conformance criteria describe procedures for testing the Definition of Conformance. Each Conformance Requirement is described in detail in Tables 11-1 through 11-3.

Table 11-1: Conformance Requirement #1: XML Schema Conformance

#	Description
CR.1	The SDP XML Document shall be “well-formed” and “valid" as validated against an approved SDP XML Schema using two or more XML Schema Validation tools. “Well-formed” and “Valid” refer to the XML Schema 1.1
CR.1.1	The SDP Schema and documents shall meet the conformance requirements of the World Wide Web Consortium standards as described in the documents listed below: Worldwide Web Consortium (W3C). (04 February 2004). W3C Recommendation: Extensible Markup Language (XML) 1.0 (Third Edition). Available from: http://www.w3.org/TR/REC-xml/ Worldwide Web Consortium (W3C). (28 October 2004). W3C Recommendation: XML Schema Part 1: Structures (Second Edition). Available from: http://www.w3.org/TR/xmlschema-1/. Worldwide Web Consortium (W3C). (28 October 2004). W3C Recommendation: XML Schema Part 2: Datatypes (Second Edition). Available from: http://www.w3c.org/TR/xmlschema-2/. Although the current editions of the standards are cited, the most recent version of these standards will apply to the SDP.

Table 11-1: Conformance Requirement #1: XML Schema Conformance

#	Description
CR.1.2	The SDP XML Document shall be checked by more than one industry-recognized XML Schema validator. The open source validator developed by the World Wide Web Consortium may be found at http://www.w3.org/2001/03/webdata/xsv.

Table 11-2: Conformance Requirement #2: Referential Integrity Checks

#	Description
CR.2	The SDP XML Document shall be checked for valid referential integrity among unique keys and referenced keys (e.g., primary and foreign keys).
CR.2.1	The following keys (identifiers) shall be checked for uniqueness: 2.1-1. amenityID.Amenity 2.1-2. blockID.Block 2.1-3. contactListID.agencyContact (within an Agency) 2.1-4. dayType.DayType 2.1-5. depotID.Depot 2.1-6. locationID.Location 2.1-7. noteID.Note 2.1-8. organizationUnitID.OrganizationUnit 2.1-9. passAccessID.PassengerAccessComponent 2.1-10. patternID.Pattern (by routeID) 2.1-11. plantComponentID.PlantComponent (within a TransitFacility) 2.1-12. portalID.Portal 2.1-13. revisionNumber.ScheduleRevision 2.1-14. routeDepotVersion.RouteDepotVersion (routeDepotChanges : ScheduleRevision) 2.1-15. routeDirection.routeDirectionList (by Route) 2.1-16. routeGroupingID.RouteGrouping 2.1-17. routeID.Route 2.1-18. seqNo.transitPathEvent (within a Pattern) 2.1-19. seqNo.transitPointEvent (within a Pattern) 2.1-20. stopID.TransitStop 2.1-21. timepointID.Timepoint 2.1-22. trackNo.Track 2.1-23. transferClusterName.TransferCluster 2.1-24. transitFacilityID.TransitFacility 2.1-25. transPathID.TransitPath 2.1-26. tripID.Trip (by routeID)
CR.2.2	Key references (Keyref) shall be checked for existence against their key when the element with the key is present. The following key and keyref pairs shall be checked: 2.2-1. key: amenityID.Amenity* (*when element is present) a. Keyref: componentID.amenityID.plantComponentList.TransitFacility 2.2-2. key: depotID.Depot*(*when element is present) a. KeyRef: depotID.routeDepotVersion.routeDepotChanges.ScheduleRevision 2.2-3. key: locationID.Location a. Keyref: locationID.Depot b. Keyref: locationBegin.Trip c. Keyref: locationEnd.Trip d. Keyref: locationID.tripTimeList.Trip

Table 11-2: Conformance Requirement #2: Referential Integrity Checks

#	Description
	e. Keyref: locationBegin.Block f. Keyref: locationEnd.Block g. Keyref: locationID.blockTimeList.Block h. Keyref: origin.Pattern i. Keyref: destination.Pattern j. Keyref: locationID.eventList.transitPointEvent.Pattern k. Keyref: origin.TransitPath l. Keyref: destination.TransitPath m. Keyref: physicalPoints.TransitPath n. Keyref: generalizeLocation.Location o. Keyref: location.Timepoint p. Keyref: locationID.TransferCluster q. Keyref: clusterLocations.TransferCluster r. Keyref: locationID.Amenity s. Keyref: locationID.PassengerAccessComponent t. Keyref: locationID.Portal u. Keyref: locationID.TransitStop v. Keyref: locationID.TransitFacility 2.2-4. key: noteID.Note a. Keyref: noteList.Note b. Keyref: notes.tripTimeList.Trip 2.2-5. key: organizationUnitID.OrganizationUnit*(*when element is present) a. Keyref: organizatonID.routeDepotChanges.ScheduleRevision 2.2-6. key: passAccessID.PassengerAccessComponent*(*when element is present) a. Keyref: componentID.passAccessID.plantComponentList.TransitFacility 2.2-7. key: patternID.Pattern (by routeID) a. Keyref: patternID.Trip b. Keyref: patternID.patternSet.groupSetList.RouteGrouping 2.2-8. key: portalID.Portal*(*when element is present) a. Keyref: componentID.portalID.plantComponentList.TransitFacility 2.2-9. key: routeDirection (by Route).routeDirectionList a. Keyref: routeDirection.Pattern 2.2-10. key: routeID.Route a. Keyref: routeID.routeDepotVersion.routeDepotChanges.ScheduleRevision b. Keyref: routeID.Pattern c. Keyref: routeID.Trip d. Keyref: routeID.EventConnection 2.2-11. key: stopID.TransitStop a. Keyref: platformNo.tripTimeList.Trip [if this is a stopID] b. Keyref: stopID.transitPointEvent.eventList.Pattern c. Keyref: componentID.stopID.plantComponentList.TransitFacility 2.2-12. key: trackNo.Track*(*when element is present) a. Keyref: trackNo.transitPointEvent.eventList.Pattern 2.2-13. key: transferClusterName.TransferCluster a. Keyref: transferClusterName.EventConnection 2.2-14. key: transitFacilityID.TransitFacility a. Keyref: transitFacilityID.Depot b. Keyref: partOf.TransitFacility

Table 11-2: Conformance Requirement #2: Referential Integrity Checks

#	Description
	2.2-15. transPathID.TransitPath a. Keyref: transPathID.Track b. Keyref: tranPathID.eventList.transitPathEvent.Pattern 2.2-16. tripID (by routeID) .Trip a. tripID.tripSet.groupSetList.RouteGrouping b. tripID.EventConnection
CR.2.3	The following dates shall be checked for inclusion within the schedule version date: Activation/deactivation dates: RouteDepotVersion → ScheduleRevision → ScheduleVersion (The route depot version activation/deactivation dates should fall within the schedule revision activation/deactivation dates which in turn should fall within the schedule version activation/deactivation dates.)

Table 11-3: Conformance Requirement #3: Document Naming Requirements

#	Description
CR.3	The SDP XML Document shall conform to the SDP XML Document naming conventions.
CR.3.1	The following naming convention should be used for a complete scheduled contained in the SDP XML Document: SDP_<agencyAcronym>_v<scheduleVersionID>.xml Example: SDP_LIB_v107.xml Where content of bracketed variables match the values of the elements within the SDP XML Document.
CR.3.2	The following naming convention should be used for a partial schedule contained in the SDP XML Document: SDP_< agencyAcronym >_<subset>_v< scheduleVersionID >.xml Example: SDP_NYCT_M1-M10_v01-04-07.xml Where the content of bracketed variables match the values of the elements within the SDP XML Document.

Test Methods

Test Methods describe the tests and procedures for ensuring SDP XML Document conformance to the specification. Several applications and validators are currently in place to test the SDP document. These include a SDP Checker application and the XML validator (see Part 3 SDP Guidance document for additional information on the SDP Checker application).

In the future, NYSDOT will implement a registration and testing web site/portal for transit service information. Additional testing methods and procedures will be specified at a later time.

Section 11.2: Conformant Profile Development

In This Section

- Learn what constitutes a conformant SDP Profile.
- Learn what SDP profiles are currently defined.
- Learn about how to create a conformant SDP Profile for a downstream application.
- Understand the issues related to extending or constraining the SDP XML Schema.

"Profile" Definition
A collection of one or more standards with applicable options, parameters and constraints within the standards, and if necessary, additional nonstandard functionalities needed to provide or support a particular function, application or platform. [ISO/IEC TR 10000-1:1998]

What is a Conformant SDP Profile?

A conformant SDP Profile identifies the options available for a particular use of a "standard" according the ISO/IEC TR 100000-1:1998 standard on developing "profiles" (see "Profile" Definition box). A profile documents how the requirements of a specification or standard are implemented. In the case of the SDP, the Conformance Situation specifies how an agency may restrict or augment the SDP. The Conformance Criteria describes the types of tests that must be passed. If the added or relaxed requirements of the SDP profile can meet those conditions, then it may become a "conformant" SDP Profile.

In summary, the SDP XML Schema may be changed so long as it follows the *Conformance Situation* described in Section 11.1. The conformance requirements, as stated in Section 11.1 *Conformance Criteria* will not change, although additional referential integrity and quality checks may be added. Furthermore, naming requirements may be added following approval by the organization(s) requesting the information.

Current Conformant SDP Profiles

The region has not yet adopted a governance process by which profiles are defined as conformant or approved. However, the specification for the SDP XML Schema Version 1.0 associated with this Guidance Document is the baseline set of requirements for the SDP. Several applications use or read the SDP Document. In particular, the regional trip planning application, TRIPS123, may soon be using the SDP Document to process all new data sets. In addition, the TSDEA project staff developed prototype interfaces to the SDP XML Document structure to drive the TriMet Timetable Publisher application (TTPub). Each of these downstream applications requires similar but different information to work properly.

The SDP XML Schema is found in Part 3 SDP Guidance document.

Appendix A: Special Considerations for Rail Transit

In This Document

- Learn how to apply the Schedule Data Profile to Rail Transit;
- Understand the special features identified in the SDP that support Commuter Rail and Subway services.

Photos from NY MTA website (9/19/2007)

Rail Concepts in the Schedule Data Profile

Although the SDP primarily draws on terminology used by bus operators, the Schedule Data Profile applies to multiple modes of public transit including commuter rail. However, because rail has unique characteristics, the application of the SDP for rail is slightly different than it is for bus. This section describes an approach for modeling rail transit using the SDP.

From a rail transit perspective the SDP focuses on services for rail customers. Personnel and equipment control and operations schedules are not part of the modeling constructs. (Note: The SDP supports elements for bus equipment scheduling, and these terms, e.g., block, may be confusing to rail operators because the meaning is different for bus and rail.)

This section is divided into two parts. The *first section* discusses special features and models applied to rail, particularly commuter rail schedules and facilities. The section describes differences between scheduling rail and bus services through examples of how a rail operator may apply the elements to model its passenger assets and services. The *second section* provides a cross-reference of key SDP data concepts and their common rail equivalents in summary format. In addition, it is formatted to provide "quick search" guidance for rail operators on how to apply the SDP data concepts to rail equivalents.

A-1: Special Features for Rail

Although rail and bus schedules are used by patrons in a similar way, schedule generation is different between the two modes, and rail facilities are more complex than bus stops and even most bus terminals. Applying the complex elements of rail to the SDP Transit Facility data concept is discussed in Section 8 on Transit Facilities Data Concept. This section focuses on rail schedule characteristics specifically Route, Service and Transit Path data concepts.

Rail schedules are driven by the constraints imposed by the rail alignment while the SDP models customer-focused schedules. Another difference between rail and bus is that a rail "vehicle" is composed of multiple cars with passenger portals on both sides of the conveyance. These characteristics require additional fields, as well as require different interpretations of other elements within SDP service and network branches. Each related element will be described below. Note that child and embedded elements are indented to show the hierarchy of the SDP XML Schema organization.

<u>Agency Registration Branch Elements</u>
The AgencyRegistration Branch includes one data concept that is impacted by Rail Transit-Route. Issues related to the Route element and its child "mode" are discussed in this section.

Route: The route is a transit convention used to group similar service together. In the case of rail, some agencies group service by destination or along a line or branch. To that end, the same trip or "train" may appear in several Public Timetables because the service stops before the service branches to the destination points. For example, LIRR may include Train 2702 in both its Montauk and Port Jefferson branch timetables. One may consider each train its own route-pattern-trip tuple. Unfortunately, merging those three concepts into one is not practical when a rail system provides several hundred service options.

> *Data Concept Approach:* Based on the SDP functional requirements, a trip or train has a unique relationship with a route. This is a relationship that cannot be adapted or broken. The SDP requirements meet customer service needs. To that end, rail operators should use Route to describe service schedules. Each trip must still be related to a single Route, indexed by an identifier and denoted with a publically recognized name. The SDP RouteGrouping concept (see Section 10) should be used to associate a trip to multiple routes, for example, when building public timetables that associate several trains with multiple branches or lines.
>
> *Mode:* The Route element contains a field for "mode". The enumeration values used for Mode come from the National Transit Database (NTD) mode list. The typical designation is "HR" (heavy rail) for subway and "CR" (commuter rail) for commuter rail. An example of a commuter rail enumeration is included in the Long Island Rail Road XML document excerpt below.

XML Excerpt of Route (example #1 from LIRR)

```
<Route>
  <routeID>DUNOBY</routeID>
  <routeName>Oyster Bay</routeName>
  <mode>CR</mode>
  …
</Route>
```

An example of a subway mode is listed below:

XML Excerpt of Route (example #2 from NYCT Rail)

```
<Route>
  <routeID>4</routeID>
  <routeName>Line 4</routeName>
  <mode>HR</mode>
 …
</Route>
```

The mode is designated on a route by route basis because some agencies provide multi-modal services.

Service Branch Elements

The Service Branch includes one major data concept that is impacted by Rail Transit- Trip. Issues related to the Trip element and several children including "tripTimeList" are discussed in this section.

Trip: The trip is defined as "A one way scheduled movement of a transit vehicle between starting and ending locations. Each trip is an instance of a pattern where service is provided for a route in a given direction." The "train" is similar to a trip. The tripName was inserted as a child element to specifically support a rail concept. The XML Excerpt for Trip is from the Metro North Railroad XML document.

tripName: When the tripID is not the same as the train number, the tripName should be used as the primary train number. For example, LIRR uses the primary train number as the tripName while the tripID is a unique sequential index. In the MNR example below, the primary train number is used for both the tripID and the tripName.

XML Excerpt of Trip (from MNR)

```
<Trip>
  <tripID>1231</tripID>
  <routeID>3</routeID>
  <patternID>5</patternID>
  <dayType>weekday</dayType>
  <tripName>1231</tripName>
  <tripType>1</tripType>
  <timeBegin>29520</timeBegin>
  <timeEnd>31920</timeEnd>
  <locationBegin>112</locationBegin>
  <locationEnd>1</locationEnd>
      . . .
</Trip>
```

There are other child elements that are applied in a slightly different way to support rail concepts. These include the locationBegin and locationEnd pair, and tripTimeList (TripTime data concept).

locationBegin (locationEnd): The begin and end location of the trip should designate the origin station to the destination station. The general guidelines specify that the locationBegin and locationEnd should exactly match the events included in the pattern. The application of this business rule is relaxed for rail. The issues related to matching trip begin/end to pattern origin/destination pairs are discussed in more detail below under the Transit Network Branch Elements section.

*tripTimeList (embedded set of **TripTime** elements)*: The tripTimeList is an ordered set of TripTime elements. The TripTime element is the point at which service is measured. In the case of transit, each TripTime is a point where service is delivered to the customer as at a stop at station. Because operations and crewing are not part of the SDP, the trip timing points should include only station stops where patrons may alight or board the train. There are other concepts embedded in the TripTime element that specifically supports customers navigating the rail service.

timeType: The timeType defines the event as the beginTrip, endTrip, arrival or departure time.

tripEventType: The tripEventType may identify key information about service at a stop such as side or cars where boarding/alighting are allowed. Some information may be provided as a note if no downstream application uses the information in an algorithm. An example of this is shown in the excerpt below where <notes> 2</notes> may be referenced as: "All patrons alighting from the train at South Ferry should board one of the first four cars due to platform construction." In this example, a trip planning application may attach the note to the trip plan. If the boarding information triggered an alert to the rider when he boarded the train, then the information would need to be specified as a tripEventType. See Section 5 for more information on creating a Note reference.

XML Excerpt of tripTimeList (from NYCT Rail)

```
<tripTimeList>
        <tripTime>3600</tripTime>
        <tripEventType>AlightBoard</tripEventType>
        <timeType>arrival</timeType>
        <locationID>140</locationID>
        <platformNo>0A</platformNo>
        <notes>2</notes>
</tripTimeList>
```

platformNo: The platformNo element was inserted to provide more detailed information to the rail patron on the location of the boarding/alighting location; (the element may also be used to designate a bus bay). In the excerpt above, the platformNo is 0A.

seqNo: Although the sequence number is not needed to order the tripTimes (the time value is an ascending integer), a seqNo should still be inserted that matches the Pattern event. Specifically, the TripTime seqNo should match the associated event seqNo (TransitEventPoint.seqNo) found in the Pattern eventList. The Pattern eventList, as discussed in the Transit Network Branch Elements, catalogs all the stops for each unique branch (origin-destination pair and all intermediate stops irrespective of skipped stops). The rationale for ensuring the mapping between the Trip times and Pattern events relates to the needs of a few applications that build an unambiguous mapping between the two set of points. When the Pattern eventList exactly matches the associated Trip tripTimeList, then matching the seqNo is not necessary.

For example, in the MNR XML excerpts below, the Trip timePointList seqNo matches the Pattern eventList seqNo for the routeID 3, patternID 5.

XML Excerpt of Matching seqNos in tripTimeList (embedded in Trip) and transitPointEvent (embedded in Pattern) (from MNR)

```xml
<tripTimeList>
    <tripTime>25620</tripTime>
    <timeType>beginTrip</timeType>
    <locationID>108</locationID>
    <platformNo>3</platformNo>
    <seqNo>108000</seqNo>
    <notes>S</notes>
</tripTimeList>

<transitPointEvent>
     <locationID>108</locationID>
     <seqNo>108000</seqNo>
     <ptEventType>1</ptEventType
</transitPointEvent>
```

<u>Transit Network Branch Elements</u>

The Transit Network Branch includes two data concepts, Pattern and Transit Path, both impact Rail Transit. Issues related to the Pattern element and several children including "eventList" are discussed in this section. In addition, how the TransitPath may be used to describe the rail alignment or physical track descriptions used by rail is discussed in this section.

Pattern: Service for many commuter rail as well as commuter bus operators differs for every trip. Look across a timetable of stops on a line and you will find few similar trips, not like the frequency of service offered by a local or regional bus routes. Conceptually, a one-to-one relationship between pattern and trip (train) exists. So there are two ways to handle Pattern.

> *Data Concept Approach #1:* Create a unique Pattern for every Train so there is a one-to-one relationship between Pattern and Train (Trip). Although a simple solution, this makes for a larger file of potentially redundant information.
>
> *Data Concept Approach #2:* Create a unique Pattern for every Route (branch or line) by Route Direction (for every non-branching origin-destination pair). The sequence numbers for each event in the Pattern eventList should then match the Trip tripTimeList records. The match will directly associate each Trip tripTime to the specific stop along the Pattern. As a consequence, the Trip tripTimes are mapped to specific Pattern events without ambiguity if stops are skipped or the Trip is implemented as a short turn. The rules to ensure that there is an unambiguous mapping between Trip "trip times" and Pattern events is discussed in the Trip section above.

When service is frequent with few variations and many recurrent trips, such as subway service, then the Pattern description should follow the same convention as bus service.

origin – destination: Similar to Trip's locationBegin – locationEnd pair, the origin – destination pair should be the first and last station in the Pattern. Non-revenue locations should not be included unless patrons (or crew) board/alight at those locations.

eventList (embedded set of TransitPointEvent elements): The eventList is an embedded set of one or more TransitPointEvent elements. The eventList is a linearly ordered set of locations that describe the physical path and events that occur at that point along the Pattern. In the case of rail, the path is "logical," not the actual track over which the train travels. Although any event type may be included, the customer-focused purpose of the eventList is to include the locations where passengers may board or alight the train.

trackNo: The trackNo is the track known by the public at the station, not necessarily the track index or identifier used internally to manage track assets.

stopID: This element should be used to identify the platform, if known.

distanceFromOrigin: This element refers to the Pattern origin element.

ptEventType [eventType_cd]: Similar to tripTimeType, this enumerated type is used to specify local event types. The SDP XML Schema does not currently list enumerated type values except for "1" timing point and "2" other. To that end, new types may be added as a unique numeric value set by local users. The value of the local code set should be submitted with the XML document (see *SDP Guidance Document Part 3)*. More than one ptEventType may be included in the XML Document. In the invented XML excerpt below, the last station on Line 1, the South Ferry station (locationID = 37), opens its doors on the right hand side (ptEventType = 4), and has a shortened platform due to construction (ptEventType value "15" indicates that the platform is shortened). The "5" indicates that all patrons alighting at South Ferry should be in one of the first four cars. This enables downstream applications to use these values to logically process the information.

XML Excerpt of transitPointEvent (adapted from NYCT Rail)

```
<transitPointEvent>
 <locationID>37</locationID>
 <seqNo>37000</seqNo>
 <ptEventType>1</ptEventType>
 <ptEventType>4</ptEventType >
 <ptEventType>15</ptEventType>
</transitPointEvent>
```

TransitPath: The TransitPath element is used to describe transit network elements that may not be described by a transportation network model (such as a map database). Tracks, paths through parking lots, depots and transit only lanes are not typically represented in commercial or even public-funded transportation networks. The TransitPath may also be used to describe paths that are associated with operational performance significance, for example, peak load segment, running time segments, route segments that are the building blocks for Patterns. For rail, the TransitPath may be used to represent the physical description of the track network. The representation is consistent with several spatial data formats.

Additional information on TransitPath guidance may be found in Section 6.2. Note, the TransitPath may also be used to "patch" a map database when "private" or "non-commercial" streets and alignments are not included.

A-2: Quick Search Guidance for Rail SDP

The quick search guide is a table that lists the SDP Data Concept, its Definition and guidance to rail operators on how to apply the concept to rail characteristics. Table A-1 SDP Data Concept Rail Guidance contains three columns:

- SDP Data Concept and the SDP XML Schema branch in which it is positioned,
- Formal data concept definition, and
- Guidance on how to apply the concept to rail

The table includes all the data concepts that are found in the SDP XML Schema. Most data concepts are applied in a similar way to bus and rail. When this is the case, the Guidance field states: "Same for all modes".

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
AdditionalContact (Agency Registration)	Agency contact information that is in addition to the information supplied in the Agency element.	Same for all modes
Agency (Agency Registration)	"A transit agency is an organization that provides transportation services by bus, rail, or other conveyance to the general public or special services on a regular, continuous basis."	Same for all modes
Amenity (Transit Facility)	**Elements of a physical feature, a fixed location, or a transit facility.** Example: the amenities of a public transportation stop may include the shelter, platform announcement panel, and benches. Note: an amenity may be described by one or more characteristics, or attributes, such	Same for all modes. The amenity may be used as a template or generic feature (such as a trash can). An instance of a Transit Facility may then include it as one of many Plant Components.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
	as the year of construction or its current condition. [GOS, part 7d, p., 2]	
Block (Service)	The daily, sequence of revenue and non-revenue trips assigned to a transit vehicle in revenue service from pull-out to pull-in.”	**Not applicable** to Rail Services. A Block relates to scheduling buses.
BlockTime (Service)	The path on which the vehicle travels is described by the Block Times, a series of coordinates, also linear references (distanceFromStart) from the beginLocation. The Block Event Times (blockTimes) are an ordered set. Each blockTime is associated with a specific event along the block. Each Block Event Time is associated with one trip. The Block Event Time may be any type of transit feature type: timepoint, transit stop or other event (e.g., fare set change, headsign change) that is referenced by a Location.	**Not applicable** to Rail Services. A Block relates to scheduling buses.
ConnectionSegment (Transfer Cluster)	The directions between two stops within a Transfer Cluster. The directions may describe accessible and mobility-challenged instructions.	For coordinated rail to rail, rail to bus, and generally mode to mode transfers, this construct provides directions from one trip (or train) to another.
DayType (Agency Registration)	A classification for the service provided on a type of day or day of the week. Among the examples of day type include: Monday, weekday, Saturday, holiday, Thanksgiving, Wed before Thanksgiving, New Years Eve, New Years Day.	Same for all modes
Depot (Agency Registration)	A storage facility operated by a transit agency or its agent where transit vehicles park overnight and are maintained. alias: base, garage, division, yard	Same for all modes. Detailed descriptions of facilities within the depot are not within the scope of the SDP.
EventConnection (Service)	An entity that describes an agency specified connection or transfer between two trips on different routes.	This element may provide a connection between two trains on the same branch or a coordinated transfer between train and bus services.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
Location (Gazetteer)	This entity represents a place in the Transit Gazetteer. It contains the location description for points used to describe or relate a transit network over which transit service is provided.	The location of a place related to train service. The location may describe the path of the tracks (if used in TransitPath), or the location of a station. In most cases this feature describes the location of the rail station or one of its portals (entrances).
Note (Service)	A note for the public that describes a characteristic of the service. The note may be associated with a Trip or a Trip Time.	Same for all modes. May be attached to a Trip or tripTimeList (TripTime) element.
Organization (Agency Registration)	A part of an Agency that issues or uses a transit schedule.	Same for all modes.
PassengerAccessComponent (Transit Facility)	The components used to aid travelers to traverse from one level to another or from one end of a facility to another. Examples include stairs, elevator, escalator, moving walkway. The component may be described by direction (up, down, or both), accessibility for people with disabilities or carts, and other characteristics.	This element is used to represent accessible paths in a rail station or between the rail station and another transit facility or stop.
Pattern (Transit Network)	A unique, non-branching, ordered sequence of transit paths, time points, or transit stops to be followed by a transit vehicle in scheduled service for a route in a given direction.	A pattern is not a typical modeling object in rail service. However, many downstream applications require the one way path from origin to destination of all *transit* service. Rail service may provide a one-way, non-branching, sequential list of each set of stations along its branch lines in lieu of representing all unique patterns. There are some rules with respect to matching the seqNo of each Pattern eventList element with the seqNo of each Trip tripTimeList element.
PlantComponent (Transit Facility)	A Plant Component is a physical part of a larger facility such as a boarding area, turnstile, fare vending machine, information booth, escalator, stairs, etc. The five specific types of plant components included in this model are Track, Transit Stop or	Same for all modes although rail stations are typically more complex than bus stops because it may include platforms, tracks, ticket vending equipment, gates, stairs, portals, parking (park & ride lots), etc.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
	boarding area, Amenity, Access Component and Portal.	
Portal (Transit Facility)	A place where transit customers may enter or exit a transit facility, station or stop. Examples include doors and gates to transit facilities.	Same for all modes.
RelativeLocation (Location)	A linear reference or attribute related to a Location. Fields include position relative to intersection, off or on transportation network, etc.	RelativeLocation describes the relationship of the bus stop to the road network. This element does not tend to apply to Rail.
Route (Agency Registration)	A collection of patterns and associated trips in revenue service with a common identifier or name. Rail alias: Line or Branch	The Route is a collection of services that are grouped together. A Route has a one-to-many relationship with trips. A trip is associated with only one route. Use RouteGrouping for timetable generation.
RouteDepotVersion (Agency Registration)	Characteristics of the schedule version that applies to a route that operates from a specific depot or that is generated by a specific organizational unit. In the case where there is no distinction on specific organizational unit, the depotID or organizationUnitID is a non-identifying foreign key.	Same for all modes. This may be used for updates to select services or Route (Branch or Line) changes.
RouteDirection (Route)	The direction of travel of a transit conveyance along the physical path. The element allows for public facing information.	Same for all modes, there is a first and second direction for most routes. The publicRouteDirection may use a destination location as the route direction, e.g., “Babylon”.
RouteGrouping (Agency Registration)	The Route Grouping entity describes a collection of Patterns or Trips that are grouped together for a particular purpose (i.e., route grouping type: A set of patterns or trips may be grouped together operationally in order to closely coordinate their scheduled headways along a common alignment or carriageway. Public timetables group routes together to communicate the frequency of service of routes that share a common corridor before they branch).	This concept may be used to group trips into more than one branch for the purpose of publishing the information in a timetable.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
ScheduleRevision (Agency Registration)	Records the state and manages the changes to the designated schedule version.	Same for all modes.
ScheduleVersion (SDP XML Document header attribute group)	The time period, described by the start date and time and optionally end date and time, when an agency service provision is valid. Due to the nature of transit schedules, different routes, depots and organizational units may implement various versions that operate during the same time period. Example: Long Island Bus typically issues a new Schedule Version in March, June, September and January.	Same for all modes.
Status (Plant Component)	The state or condition of the Plant Component with respect to the dates of placement, activation, deactivation, etc.	Same for all modes.
Timepoint (Gazetteer)	A location at which time is measured for a trip.	Rail only needs timepoints at rail stations.
TimetableHeader (Route)	The header row used for a route timetable.	Same for all modes. The timetable is for use by the public. A "crew" timetable was not part of the downstream applications that generated the functional requirements.
Track (Transit Facility)	"A pair of parallel rails, and required ties and fastenings, over which trains move." [LIRR] Also known as Lane and Berth	The track referenced at a station. Track may also be monorail.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
TrackAssociation (Transit Facility)	An association between a specific platform and track. A Platform may be associated with multiple Tracks, for example, platform "A" at Jamaica Station is flanked on both sides by track 1 and 2. Alternatively, a track may support multiple platforms, such as platforms A and B are served by Track 2. This entity distinguishes the combined relationship between one platform and an adjacent track.	See note in Definition column.
TransferCluster (Gazetteer)	A place where transit customers may transfer from one service provision to another such as from one bus to another, or from one mode of travel to another. The Transfer Cluster may include related information on the connecting trips (see EventConnection) and directions between the connecting stops (ConnectionSeg).	Same for all modes.
TransitFacility (TransitFacility)	A building or center used by a transit vehicle or transit operator for the purpose of parking, storing, maintaining or providing services to transit customers. The SDP uses this entity to represent multiple transit stops wherein transfers may occur between routes, modes and/or operators. Although a transit facility in a general sense may also encompass a vehicle garage, maintenance yard, administration building or other facility type, within the context of the SDP, only facilities that serve the public need be included.	Same for all modes. There are several detailed examples on rail and multimodal facilities in Section 9.
TransitPath (Transit Network)	A path over which transit service in provided. Alias: route segment, timepoint interval	The Transit Path element may be used to describe the physical path of the rail track.
TransitPathEvent (Pattern)	A path that composes the	The Transit Path Event is used by

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
	alignment of a pattern where service is delivered. The path should include stops, timepoints and other events.	the Pattern in place of the TransitPointEvent.
TransitPointEvent (Pattern)	A place where transit service is delivered along the transit network.	The TransitPointEvent element includes some specific elements that may be used by rail service. In particular, the ptEventType (eventType_cd) may be used to designate the location of the event relative to the platform for the purpose or providing information on boarding or alighting the train.
TransitStop (Transit Facility)	An established location where public transportation customers may board or alight a transit vehicle in revenue service. Alias: bus stop, boarding area, ramp, platform.	A transit stop is typically used to describe the platform at the station, for example, the Nostrand Ave. Station Platform A may be described as NAV-A.
Trip (Service)	A one way scheduled movement of a transit vehicle between starting and ending locations. Each trip is an instance of a pattern where service is provided for a route in a given direction. Rail alias: Train	The Train is the equivalent service concept as the Trip. Both rail and commuter bus trip times are associated with stops or stations. The tripName is specifically inserted to capture the primary train number.

Table A-1: SDP Data Concept Rail Guidance

Data Concept (Branch)	Definition	Application to Rail
TripTime (Service)	The time along a trip when a vehicle is scheduled to pass (arrives at/departs from). The trip time may be published or unpublished, coordinated or uncoordinated with trip times of other trips.	Both rail and commuter bus trip times are associated with stops or stations along a Pattern, which for rail is defined as a unique origin-destination pair and the stations along the path. The TripTime elements should comply with designated business rules. Key among them is that the seqNo should match the same location designated by the associated Pattern eventList seqNo. Additional customer tripTimeType codes or special notes may be included with trip times to describe customer directions such boarding and alighting permitted, reservations required, luggage storage available.

Appendix B: Glossary

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
T	_cd	A suffix that designates that the element is an enumerated type that includes all the code values.	
T	_id	A suffix that refers to elements that are identifiers.	
F	Amenity	A physical feature of a fixed location of a transit facility. They may include the shelter, platform announcement panel, wastebasket, phone, benches, and parking (e.g., park and ride lot).	
T	Annotation	Generally used to define data concepts or provide alternative values for enumerated types, annotations are documentations within the XML Schema that are human-readable documentation or machine-readable code to define the element.	
T	Attribute	A qualifier on an XML tag that provides additional information.	
T	Attribute Group	An attribute group qualifies an element. This feature is implemented in the SDP on the SDP root node to describe the Schedule Version.	
T	Branch Elements	A complex type element that classifies a set of nested elements in the XML Schema.	
F	CDRM	Conceptual Data Reference Model. The conceptual model describes "real-world" phenomenon using unambiguously defined set of data concepts and model their relationship to each other.	
T	Child Element or Child Entity	An element or entity that is related to another element or entity by being embedded in that entity or element.	
T	Comment	Comments begin with `<!--` and end with `-->`. Comments can contain any data except the literal string `--`. You can place comments between markup anywhere in a XML document. Comments are not part of the textual content of an XML document. An XML processor is not required to pass them along to an application.	
F	Complete Submittal	A SDP document submitted by a Transit Provider that includes their schedule which includes all routes in service.	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
T	Complex Type Element	An element that aggregates simple type elements or constrains other elements. See Element.	
F	Conceptual Model	See Model, Conceptual.	
F	Data Concept	"Any of a group…referring to abstractions or things in the natural world that can be identified with explicit boundaries and meaning, and whose properties and behavior all follow the same rules."	IEEE 1489:1999
F	Data Model	"A data model is a model that describes in an abstract way how data is represented in a business organization, an information system or a database management system."	wikipedia.org
F	Data Model, Conceptual	"Description of a real world domain in terms of entities, relationships and attributes, in an implementation independent manner. It should provide a structure on which the rest of the development of an application system can be based."	TRANSMODEL version 5.1
F	Data Model, Logical	"Data design, that takes into account the type of database to be used, but does not consider means of utilization of space or access"	TRANSMODEL version 5.1
T	Element	A data concept that is included in an XML Schema. An element description may be defined as a *simple type* element which describes a field or record in the XML schema, or a *complex type* which includes multiple simple or complex type elements.	
F	Entity	(1) Anything of interest (such as a person, place, process, property, object, concept, association, state or event) within a given domain of discourse. (2) an object (data) that has its own existence (as opposed to an attribute).	(1) IEEE 1489:1999 (2) TRANSMODEL version 5.1
F	Entity-Relationship Diagram (ERD)	A representation of a methodology allowing for the representation of a data model in terms of entities and relationships.	Adapted from TRANSMODEL version 5.1
T	Extensible Markup Language	See XML	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
F	Feature	A term used in the geodata industry to mean a place that is spatially located or related to a physical location. SDP features may include, transit stop, transit facility, timepoint, transfer cluster, physical point, landmark and more.	
F	General Agency Information	A layer in the SDP CDRM that describes the Agency, Schedule Version and high level Route information among other data concepts. The General Agency Information layer corresponds to the AgencyRegistration Branch in the SDP XML Schema.	
T	Geocoding	Geocoding is the process of assigning geographic identifiers (e.g., codes or geographic coordinates expressed as latitude-longitude) to map features and other data records, such as street addresses.	Wikipedia.org
T	GML – Geography Markup Language	The XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML standards consists of several documents that describe geography and geo spatial phenomenon (relating to standards).	
F	Logical Model	See Data Model, Logical	
T	maxOccurs	Attributes of an element declaration that indicates the number of times the element may be repeated.	
T	minOccurs	Attributes of an element declaration that indicates that the element is optional (the minimum number of elements required is zero).	
T	Namespace	A declaration made in the XML Schema as a means to avoid naming conflicts among different community standards referenced by the XML Schema.	
T	Namespace Reference	A prefix to the type name that distinguishes SDP defined types and native XML types, as in XML schemas. For example, “xsd” is the used as the namespace reference for the base XML Schema standard.	
T	Nested Element	A complex type element that is embedded in another complex type element.	
F	Notes	A key source of information used to explain special information to the user. The drawback of using notes is that they must be manually inserted and may not be well maintained over a long period of time.	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
F	Passenger Access Component	A component characterized by whether the access component is stairs, moving walkway, elevator, escalator or other.	
F	Pattern	A directed path (from origin to destination) of a route, that is, composed of events or event segments.	
F	Data Model, Physical	The data model that is specially designed to implement a specific database engine, for example, Oracle 9i. The physical data model may be implemented in a different format than a logical or conceptual model so as to increase operational efficiency. To that end, the physical model may duplicate data in related tables, join two or more tables so that downstream applications do not need to execute queries, may generate numeric indices on primary key(s) for faster lookup and querying.	
F	Plant Component	A constituent part of the transit facility.	
T	Reference Keys	A key used to ensure the uniqueness of the key throughout a SDP document.	
F	Referential Integrity	Referential integrity in a relational database is consistency between coupled tables. Referential integrity is usually enforced by the combination of a primary key and a foreign key. Primary key: In database design, a primary key is a value that can be used to identify a unique row in a table. Attributes are associated with it. Examples are names in a telephone book (to look up telephone numbers) and words in a dictionary (to look up definitions). Foreign key: A foreign key (FK) is a field or group of fields in a database record that point to a key field or group of fields forming a key of another database record in some (usually different) table. Usually a foreign key in one table refers to the primary key (PK) of another table.	wikipedia.org
T	Root	The main element, message or document reference that is the content of the schema.	
F	Schedule Revision	Catalogs the version (Route Depot Version) of specified route(s) (or route(s) scheduled by a depot).	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
F	SCD	Schedule Calendar Date Profile. A related XML Schema that describes a calendar wherein trips may be assigned to operate on specific calendar dates.	
F	SDP	Schedule Data Profile. The Schedule Data Profile (SDP) is a specification that describes operator generated schedule and related data. It is a business semantics specification that describes schedule information, specifically each data element and its relationship to scheduling data concepts, and preserves the referential integrity of these data concepts. The SDP description is based on recognized information technology (IT) standards such as Extensible Markup Language (XML) and XML Schema, as well as standards and best practices in the IT and transit industries.	
T	SDP Guidance Template Worksheet (“Cheat Sheet”)	An Excel worksheet that may be used for mapping native data formats to the SDP. The worksheets include Code definitions, Element definitions, and descriptions of each element in the SDP XML Schema.	
F	SDP XML Document	Data organized in a format that conforms to a prescribed XML Schema. May also be referred to as a ‘Content File’ or ‘Instance.’	
F	Service Provision	A layer in the SDP CDRM that describes the service provided (in the form of Trips, Notes, Blocks and Connections).	
T	Simple Type Element	An Element that does not have element children or attributes. See Element description.	
F	Subset Submittal	A subset of a complete schedule that includes a specified set of routes.	
T	Tag	The semantic structure delimiting the start and end of an element e.g., < tag> data </tag>	
F	Transfer Cluster	A transit facility that is logically joins a cluster of stops for the means of specifying transfer locations and the connections between them.	
F	Transit Facility	Any building or physical location managed by a transit operator or location where the public may access fixed route transit service.	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
F	Transit Schedule Data Exchange Architecture (TSDEA)	The TSDEA is a framework for managing and exchanging schedule data through the deployment of the Transit Schedule Data Exchange Architecture. The TSDEA should be viewed as an engine that integrates regional transit data, providing consistent data across the region. It enables a scalable, modular, computing framework to deploy regional transit business services.	
F	Transit Gazetteer	A layer in the SDP CDRM that inventories the "features" owned, operated or used in the transit network or service.	
F	Transit Network	A layer in the SDP CDRM that describes a logical description of the physical network over which service is provided.	
F	Valid Submission	Checking of a registered document to ensure that conforms to the requirements of the SDP XML Schema and other business rules outlined in the SDP Functional Requirements document. XML validation checks for correct data type conformance, data organization and mandatory elements as defined by the SDP XML Schema. The SDP business rules checks for identifier uniqueness, matching identifier references, dates within range, and other key rules.	
T	Well-Formed	A XML document that has correct XML syntax.	
T	XML	The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data.	wikipedia.org
T	XML Schema	An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntax constraints imposed by XML itself. An XML schema provides a view of the document type at a relatively high level of abstraction.	wikipedia.org
T	XML Validation	The process of checking to see if an XML document conforms to its related XML schema.	
T	XML Validator	A tool that checks that an XML document is well formed (organizationally correct) and valid (syntactically correct).	

T=technology related; F=-functional (SDP) related

Class	Term	Description	Source
T	XMLSpy	A commercial software product that enables the development and management of XML schemas and family of schemas.	

Appendix C: Resources

NYSDOT TSDEA Project Files
Concept of Operation for the Schedule Data Profile (Revised November 29, 2005)

Functional Requirements for the Schedule Data Profile (28 June 2006)

XML Schema Resources

XML Schema Tutorial
http://www.w3schools.com/schema/default.asp

A Really, Really, Really Good Introduction to XML, by Tom Myer
August 24 2005, http://www.sitepoint.com/article/really-good-introduction-xml

Introduction to XML, by Harry Fuecks, November 25, 2002,
http://www.sitepoint.com/article/introduction-xml

Appendix D: Notation for Conceptual Data Reference Model

(Extracted from the *Functional Requirements For the Schedule Data Profile* (28 June 2006))

Symbols and Abbreviations

This section describes the symbols and abbreviations used in the CDRM.

An entity-relationship diagram is used to describe the relationship among entities. The entity-relationship notation connects entities with links representing various relationships between them. The relationships below are illustrated in Figure D-1.

Relationships/Links: The crow's feet on the links symbolize "many" while the one or two bars signify "one." The open circle indicates there is a "zero-to-one" or "zero-to-many" relationship between the entities or tables. The relationship between two entities is typically described on the link, for example, Figure D-2 shows the relationship between an Agency and Organizational Unit as "an Agency *has* an Organizational Unit", as well as the relationship between an Organizational Unit and an Agency -- "an Organizational Unit *is part of* an Agency."

Entities/Tables: The tables are entities. The "pi" in the table indicates that the attribute is a primary key or primary identifier which is used to uniquely identify the entity. Mandatory attributes are indicated with an "M". Mandatory attributes must be included in the entity, while optional attributes are not required. In Figure 2, agencyID is a primary identifier (signified by the "<pi>" and underline) and agencyAcronym is a mandatory ("<M>") attribute of Agency; all other attributes in the Agency entity are optional.

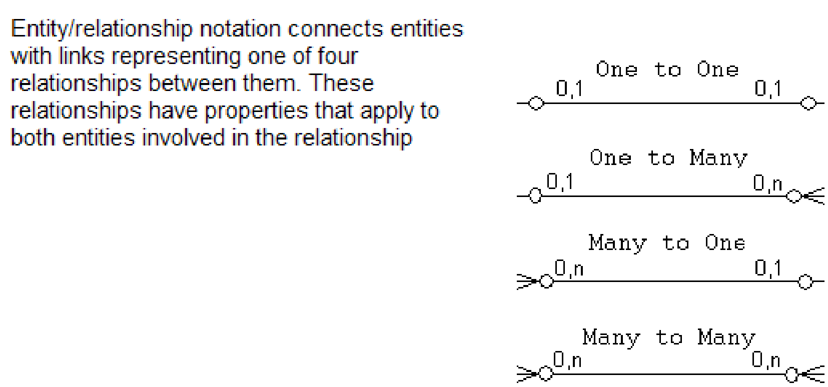


Figure D-4: Entity-Relationship Notation
[from PowerDesigner 12 CDM Users Guide, Notation]

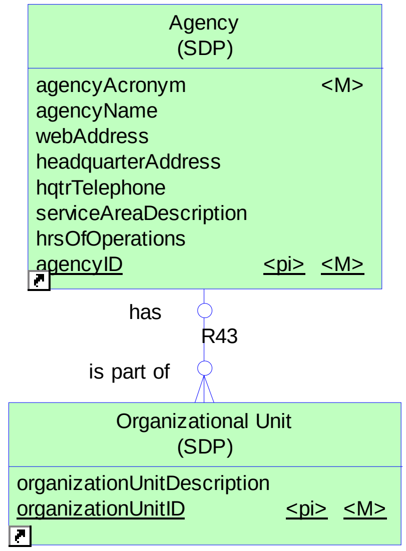


Figure D-5: Example of Entity and Relationship Diagram

Note: A conceptual data model does not explicitly identify the foreign key (or directly related primary keys). For example, in Figure 7, Organizational Unit is one of many units of a specific agency. In a physical data model, the Organizational Unit table will contain a mandatory field called "agencyID" which is a *foreign key*.

Element Naming Conventions

This section describes the naming conventions used for the CDRM data elements, concepts and entities.

An element naming convention helps a reader recognize the type of element referenced. In an entity-relationship conceptual model, there are four types of elements, three are defined in Table D-1 and described throughout this document (attribute, entity and relationship), and in addition, there is a *Domain* or reference data type name. These elements will use the following notation for their names:

Table D-1: Naming Conventions for CDRM

Attribute names	Attribute names are strung together using camel case, although they begin with a lower case letter. [attributeName]
Entity names	Entity names are camel case, strung together using underscores [Entity_Name]
Domain names	Domain names use all capital letters. (Note: in the implementation, a domain is substituted with a referenced data type). [DOMAIN]
Relationship names	No notation