WS-Calendar Platform Independent Model (PIM) Version 1.0

Committee Specification Draft 01 / Public Review Draft 01

26 April 2013

Specification URIs

This version:

http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/csprd01/ws-calendar-pim-v1.0-csprd01.pdf (Authoritative)
http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/csprd01/ws-calendar-pim-v1.0-csprd01.html
http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/csprd01/ws-calendar-pim-v1.0-csprd01.doc

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.pdf (Authoritative)
http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html
http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.doc

Technical Committee:

OASIS Web Services Calendar (WS-Calendar) TC

Chair:

Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill

Editors:

William Cox (wtcox@coxsoftwarearchitects.com), Individual
Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XMI documents for the PIM UML model: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/csprd01/xmi/

Related work:

This specification is related to:

- *WS-Calendar Version 1.0*. Latest version. http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.html.

Abstract:

The Platform Independent Model is an abstract model that will be used as the basis for determining conformance of additional models and communications with WS-Calendar and with xCal, which is based on IETF RFCs.

This is a Platform Independent Model following the Object Management Group's Model-Driven Architecture. The Platform Dependent Model from which this specification abstracts is the full model for WS-Calendar as expressed in XML (xCal).

The focus of this Platform Independent Model is on passing schedule and interval information with attachments.

Status:

This document was last revised or approved by the OASIS Web Services Calendar (WS-Calendar) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/ws-calendar/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/ws-calendar/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[WS-Calendar PIM v1.0]

WS-Calendar Platform Independent Model (PIM) Version 1.0. 26 April 2013. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/csprd01/ws-calendar-pim-v1.0-csprd01.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

Table of Figures

1 Introduction

All text is normative unless otherwise labeled.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

[ISO8601] ISO (International Organization for Standardization). *Representations of dates and times, third edition*, December 2004, (ISO 8601:2004)

[RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, http://www.ietf.org/rfc/rfc3986.txt, IETF RFC 3986, January 2005.

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[RFC5545] B. Desruisseaux *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, http://www.ietf.org/rfc/rfc5545.txt, IETF RFC5545, proposed standard, September 2009

[WS-Calendar] *WS-Calendar Version 1.0*, 30 July 2011, OASIS Committee Specification. http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html (PDF is authoritative)

[xCal] C. Daboo, M Douglass, S Lees, *xCal: The XML format for iCalendar*, http://tools.ietf.org/html/rfc6321, IETF RFC 6321, August 2011.

[XMI] *XMI Version 2.1*, September 2005, Object Management Group, http://www.omg.org/spec/XMI/2.1/ [1]

1.3 Non-Normative References

[EnergyInteroperation] *OASIS Energy Interoperation 1.0*, OASIS Committee Specification 02, 18 February 2012, http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html

[IANA] The Internet Assigned Numbers Authority, http://www.iana.org.

[MDA-Overview] *The Architecture of Choice for a Changing World*, Object Management Group, http://www.omg.org/mda/

[MDA] *OMG Model Driven Architecture Specifications*, Object Management Group, http://www.omg.org/mda/specs.htm

[Vavailability] C. Daboo, B. Douglass, Calendar Availability, http://tools.ietf.org/html/draft-daboo-calendar-availability-03, IETF Internet Draft Version 03, September 27, 2012

[UML] *Unified Modeling Language*, Object Management Group, http://uml.org/

[EnterpriseArchitect] ***Sparx*** *Enterprise Architect 9.3 and 10.0, used to produce diagrams, EAP and **[XMI]** version 2.1 files*

[1] The version of XMI as of this Working Draft is 2.4.1, but the tools used for UML support version 2.1.

1.4 Namespace

There are no XML namespaces defined in this specification.

1.5 Naming Conventions

This specification follows a set of naming conventions for artifacts defined by the specification, as follows:

For the names of attributes in UML definitions the names follow the lower camelCase convention, with all names starting with a lower case letter. For example, an attribute name might be

```
component
```

The names of UML classes, the names follow the upper CamelCase convention with all names starting with an Upper case letter followed by “Type“.

```
component: ComponentType
```

1.6 Editing Conventions

For readability, UML attribute names in tables appear as separate words. The actual names are lowerCamelCase, as specified above, and do not contain spaces.

All items in the tables not marked as “optional” are mandatory.

Information in the “Specification” column of the tables is normative. Information appearing in the “Note” column is explanatory and non-normative.

All sections explicitly noted as examples are informational and are not to be considered normative.

1.7 Architectural References and Background

WS-Calendar and this WS-Calendar PIM assume incorporation into services. Accordingly it assumes a certain amount of definitions and discussion of roles, names, and interaction patterns. This document relies heavily on roles and interactions as defined in the OASIS Standard *Reference Model for Service Oriented Architecture* **[SOA-RM]**.

Service-Oriented Architecture comprises not only the services and interaction patterns, but also the information models that support those services and make the actions meaningful. The WS-Calendar PIM is such an information model for expressing schedule and time related information in a consistent manner and to permit easy transformation or adaptation into IETF iCalendar related specifications and among implementations.

1.8 Semantics in WS-Calendar and WS-Calendar PIM

WS-Calendar PIM semantics are nearly identical to those defined in Section 1.9 of **[WS-Calendar]**. The minor differences between the referenced section and those in Section 1.9 below are listed in a non-normative Appendix.

This specification shares the same semantics and terminology, and thus allows easier exchange of information across execution environments.

[WS-Calendar] defines XML and XML Schema artifacts; the terminology differs from UML, most obviously in that XML distinguishes between elements and attributes within a type, while UML uniformly uses the term *attributes*.

1.9 Semantics

Certain terms appear throughout this document, some with extensive definitions. Table 1-1 provides definitions for the convenience of the reader and reviewer. Many terms require fuller discussion than is in this section, and are discussed in greater depth in later sections. In all cases, the normative actual definition is the one in this section.

WS-Calendar terminology begins with a specialized terminology for the segments of time, and for groups of related segments of time. These terms are defined in Table 1-1 through Table 1-4 below, and are quoted from **[WS-Calendar]**.

Table 1-1: Semantics: Foundational Elements

Time Segment	Definition
Component	In iCalendar, the primary information structure is a Component. Intervals and Gluons are new Components defined in this specification.
Duration	Well-known element from iCalendar and **[XCAL]**, Duration is the length of an event scheduled using iCalendar or any of its derivatives. The **[XCAL]** duration is a data type using the string representation defined in the iCalendar duration.
Interval	The Interval is a single Duration derived from the common calendar Components as defined in iCalendar (**[RFC5545]**). An Interval is part of a Sequence. An entire Sequence can be scheduled by scheduling a single Interval in that sequence. For this reason, Intervals are defined through Duration rather than through dtStart or dtEnd.
Sequence	A Sequence is a set of Intervals with defined temporal relationships. Sequences may have gaps between Intervals, or even simultaneous activities. A Sequence is re-locatable, i.e., it does not have a specific date and time. A Sequence may consist of a single Interval. A Sequence may optionally include a Lineage. A Sequence can be scheduled multiple times through repeated reference by different Gluons. Intervals are defined through their Duration, and the schedule, dtEnd or dtStart, is applied to the Sequence as a whole.
Partition	A Partition is a set of consecutive Intervals. The Partition includes the trivial case of a single Interval. Partitions are used to define a single service or behavior that varies over time. Examples include energy prices over time and energy usage over time.
Gluon	A gluon influences the serialization of Intervals in a Sequence, though inheritance and through schedule setting. The Gluon is similar to the Interval, but has no service or schedule effects until applied to an Interval or Sequence.
Artifact	An Artifact is the information attached to, and presumably that occurs or is relevant to the time span described by an Interval. WS-Calendar uses the Artifact as a placeholder. The contents of the Artifact are not specified in WS-Calendar; rather the Artifact provides an extension base for the use of WS-Calendar in other specifications. Artifacts may inherit elements as do Intervals within a Sequence.

WS-Calendar works with groups of Intervals that have relationships between them. These relations constrain the final instantiation of a schedule-based service. Relations can control the ordering of Intervals in a Sequence. They can describe when a service can be, or is prevented from, being invoked. They establish the parameters for how information will be shared between elements using Inheritance. The terminology for these relationships is defined in Table 1-2.

Table 1-2: Semantics: Relations, Limits, and Constraints

Term	Definition
Link	The Link is used by one WS-Calendar object to reference another. A link can reference either an internal object, within the same calendar, or an external object in a remote system.

Term	Definition
Relationship	Relationships link between Components for Binding. ICalendar defines several relationships, but WS-Calendar uses only the CHILD relationship, and that only to bind Gluons to each other and to Intervals.
Temporal Relationship	Temporal Relationships extend the **[RFC5545]** Relationships to define how Intervals become a Sequence by creating an order between Intervals. The Predecessor Interval includes a Temporal Relation, which references the Successor Interval. When the start time and Duration of one Interval is known, the start time of the others can be computed through applying Temporal Relations.
Availability	Availability expresses the range of times in which an Interval or Sequence can be Scheduled. Availability often overlays or is overlaid by Busy. Availability can be Inherited.
Busy	Busy expresses the range of times in which an Interval or Sequence cannot be Scheduled. Busy often overlays Availability. Busy can be Inherited.
Child, Children	The CHILD relationship type (rel_type) defines a logical link (via URI or UID) from parent object to a child object. A Child object is the target of one or more CHILD relationships and may have one to many Parent objects.
Parent [Gluon]	A Gluon (in a Sequence) that includes a CHILD relationship parameter type (rel_type) defines a logical link (via URI or UID) from parent object to a child object. A Parent Component contains one or more CHILD Relationships

WS-Calendar describes how to modify and complete the specification of Sequences. WS-Calendar calls this process Inheritance and specifies a number of rules that govern inheritance. Table 1-3 defines the terms used to describe inheritance.

Table 1-3: Semantics: Inheritance

Term	Definition
Lineage	The ordered set of Parents that results in a given inheritance or execution context for a Sequence.
Inheritance	Parents bequeath information to Children that inherit them. If a child does not already possess that information, then it accepts the inheritance. WS-Calendar specifies rules whereby information specified in one informational object is considered present in another that is itself lacking expression of that information. This information is termed the Inheritance of that object.
Bequeath	A Parent Bequeaths attributes (Inheritance) to its Children.
Inherit	A Child Inherits attributes (Inheritance) from its Parent.
Covarying Attributes	Some attributes are inherited as a group. If any member of that group is expressed in a Child, all members of that group are deemed expressed in that Child, albeit some may be default values. These characteristics are called covarying or covariant. A parent bequeaths covarying characteristics as a group and a child accepts or refuses them as a group.
Decouplable Attributes	Antonym for Covarying Attributes. Decouplable Attributes can be inherited separately.

As Intervals are processed, as Intervals are assembled, and as inheritance is processed, the information conveyed about each element changes. When WS-Calendar is used to describe a business process or

service, it may pass through several stages in which the information is not yet complete or actionable, but is still a conforming expression of time and Sequence. Table 1-4 defines the terms used when discussing the processing or processability of Intervals and Sequences.

During the life-cycle of communications concerning Intervals, different information may be available or required. For service performance, Start Duration and the Attachment Payload must be complete. These may not be available or required during service advertisement or other pre-execution processes. Table 1-4 defines the language used to discuss how the information in an Interval is completed.

Table 1-4: Semantics: Describing Intervals

Term	Definition
Designated Interval	An Interval that is referenced by a Gluon is the Designated Interval for a Series. An Interval can be Designated and still not Anchored.
Anchored	An Interval is Anchored when it includes a Start or End, either directly or through Binding. A Sequence is Anchored when its Designated Interval is Anchored.
Unanchored	An Interval is Unanchored when it includes neither a Start nor an End, either internally, or through Binding. A Sequence is Unanchored if its Designated Interval Unanchored. *Note: a Sequence that is re-used may be Unanchored in one context even while it is Anchored in another.*
Binding	Binding is the application of information to an Interval or Gluon, information derived through Inheritance or through Temporal Assignment.
Bound Element	A Bound Element refers to an Element and its Value after Binding, e.g., a Bound Duration.
Bound Interval	A Bound Interval refers to an Interval and the values of its Elements after Binding.
Bound Sequence	A Bound Sequence refers to a Sequence and the values of its Intervals after Binding.
Partially Bound	Partially Bound refers to an Interval or a Sequence which is not yet complete following Binding, i.e., the processes cannot yet be executed.
Fully Bound	Fully Bound refers to an Interval or Sequence that is complete after Binding, i.e., the process can be unambiguously executed when Anchored.
Unbound	An Unbound Interval or Sequence is not itself complete, but must still receive inheritance to be fully specified. A Sequence or Partition is Unbound if it contains at least one Interval that is Unbound.
Constrained	An Interval is Constrained if it is not Anchored and it is bound to one or more Availability or Free/Busy elements
Temporal Assignment	Temporal Assignment determines the start times of Intervals in a Sequence through processing of their Durations and Temporal Relations.
Scheduled	A Sequence or Partition is said to be Scheduled when it is Anchored, Fully Bound, and service performance has been requested.
Unscheduled	An Interval is Unscheduled if it is not Anchored, nor is any Interval in its Sequence Anchored. A Sequence or Partition is Unscheduled if none of its Intervals, when Fully Bound, is Scheduled.

Term	Definition
Predecessor Interval	A Predecessor Interval includes a Temporal Relation which references a Successor Interval.
Successor Interval	A Successor Interval is one referred to by a Temporal Relationship in a Predecessor Interval.
Antecedent Interval(s)	Antecedents are an Interval or set of Intervals that precede a given Interval within the same Sequence
Earliest Interval	The set of Intervals at the earliest time in a given Sequence
Composed Interval	A Composed Interval is the virtual Interval specified by applying inheritance through the entire lineage and into the Sequence in accord with the inheritance rules. A Composed Interval may be Bound, Partially Bound, or Unbound.
Composed Sequence	A Composed Sequence is the virtual Sequence specified by applying inheritance through the entire lineage and into the Sequence in accord with the inheritance rules. A Composed Sequence may be Bound, Partially Bound, or Unbound.
Comparable Sequences	Two Sequences are Comparable if and only if the Composed version of each defines the same schedule.

105

2 Introduction

The Object Management Group's Model Driven Architecture **[MDA-Overview][MDA]** is a way of describing relationships between Unified Modeling Language **[UML]** models.

An instance of MDA has two classes of models:

- A *Platform-Independent Model*, abbreviated *PIM*, of which there is one
- A *Platform-Specific Model*, abbreviated *PSM*, of which there are one or more[2]

The more abstract PIM typically captures the more abstract relationships in an architecture, making the architecture more clear. The PSM is bound to a particular *platform*.

The art of establishing an instance of MDA includes defining a PIM and PSMs, which solve interesting and important and useful problems.

2.1 The WS-Calendar PSM and PIM

In this specification we define a PIM or Platform-Independent Model for the **[WS-Calendar]** extensions to IETF **[iCalendar][xCAL]** and the relevant types included in **[xCAL]**. We use "the PIM" meaning exactly "the WS-Calendar PIM" through this specification.

[iCalendar] uses a specific platform, developed over time, to express relationships, times, events, and availability. As such, the expression is very simple, but in the aggregate relatively complex and less suitable to UML expression—the several key types have sets of values and attributes associated with them in a relatively flat hierarchy.

This PIM defines the key abstractions defined in **[WS-Calendar]** in a manner that allows for understanding the nature and information model for those abstractions. In effect, we are creating a PIM with respect to which the WS-Calendar specification is a PSM. Our purpose is to create a more abstract model of the key concepts in WS-Calendar for easier use in application development and standardization.

This specification does not depend on any specific MDA tooling or environments to be useful.

2.2 Key Abstractions

In the WS-Calendar PIM we define the following in order:

- Primitive Types for date, time, and duration
- The Interval
- Payload attachment to an Interval
- Relations
- The Gluon
- Tolerance
- Availability

2.3 Expression of the PIM

The PIM is a **[UML]** model. We represent the PIM as a normative **[XMI]** serialization of the model. The model is described using **[Enterprise Architect]**. The Enterprise Architect Project file is part of this work product but is non-normative.

[2] This is pronounced as if spelled "PISM"

2.4 Structure of the PIM Model and Specification

The PIM consists of a small number of key concepts and constructs as listed in Section 2.2. These are expressed in a largely flat structure, with a sub-package only for the Availability abstractions.[3] We have not used packages for the core abstractions, but expect (See Section 5) that conforming specifications and implementations MAY claim conformance to sub-parts of the PIM, e.g. to only the Interval.

We encourage consideration and use of the entire PIM, but understand that some aspects of the abstract model may be more complex than needed to address specific problems. We consider such profiles of the PIM to (notwithstanding the discussion on complexity and PSMs above) be a Platform-Specific Model.

We generally take the exact names for abstractions in the PIM from the names in **[WS-Calendar]** to simplify implementations and mappings where needed to bridge to a specific implementation conforming to **[WS-Calendar]**. We exercise care to disambiguate terms where there are multiple fully qualified terms of the same end component name.

Many attribute values in **[xCAL]** are conformed strings in XML, that is, strings with certain defined patterns. We require the same formats for conformed strings, and record the type in this PIM as *string* to allow easy transformation between this PIM and the many PSMs.

2.5 Nature of Expression in UML

There are constraints and semantic rules and conformance that apply to a UML model defining the WS-Calendar PIM. However, the UML cardinality expressions and constraints give a flexibility of expression, and allow for determining values in fully bound class instances that are less succinct than the standardized expressions.

For example, an instance of Interval (see Figure 2 The PIM IntervalType) might have only a duration; the UML, however, lists duration as optional (cardinality 0..1). Rules in this specification show how a specific representation is to be interpreted, typically by inheriting values from elsewhere. Conceptually, the actual values depend the context.

An Interval notionally has a start time, but that also is optional in the UML model. Finally, an Interval does not have an end time (expressed in Figure 2 as dtEnd of cardinality 0. We keep the dtEnd for ease of use in PSMs and for intermediate stages of mapping into the canonical start and duration model.

These characteristics are as defined in **[WS-Calendar]** and describe an abstract Interval with at most a start time and duration. This is in contrast to some historical models that require each interval to contain a start and end time, or occasionally start, end, and duration. The added flexibility of relocatable sets or schedules comprised of Intervals and Gluons, makes the expression of such a schedule easy and reusable, thus permitting a powerful abstraction to be applied to all sorts of scheduling expressions.

[3] The Vavailability definition is in process in the IETF.

3 The Platform-Independent Model

In this section we treat in turn each component of the PIM starting with an introduction. Each subsection has an introduction, a diagram, and discussion that may include the relationship of the respective components to the rest of the PIM.

This Platform-Independent Model (PIM) **[MDA]** describes an abstraction from which the Platform-Specific Model (PSM) of **[WS-Calendar]** can be derived. The intent is twofold:

(1) To define an abstraction for calendar and schedule more in the style of web services descriptions, and
(2) To define the PIM as a model allowing easy transformation or adaptation between systems using the family of WS-Calendar specification.

3.1 Introduction

In this section we present the entire PIM together with architectural discussion. The following sections address

- Primitive types
- The Interval
- Payload attachment an Interval
- Relations
- The Gluon,
- Tolerance, and
- Availability

3.1.1 Model Diagram of the PIM

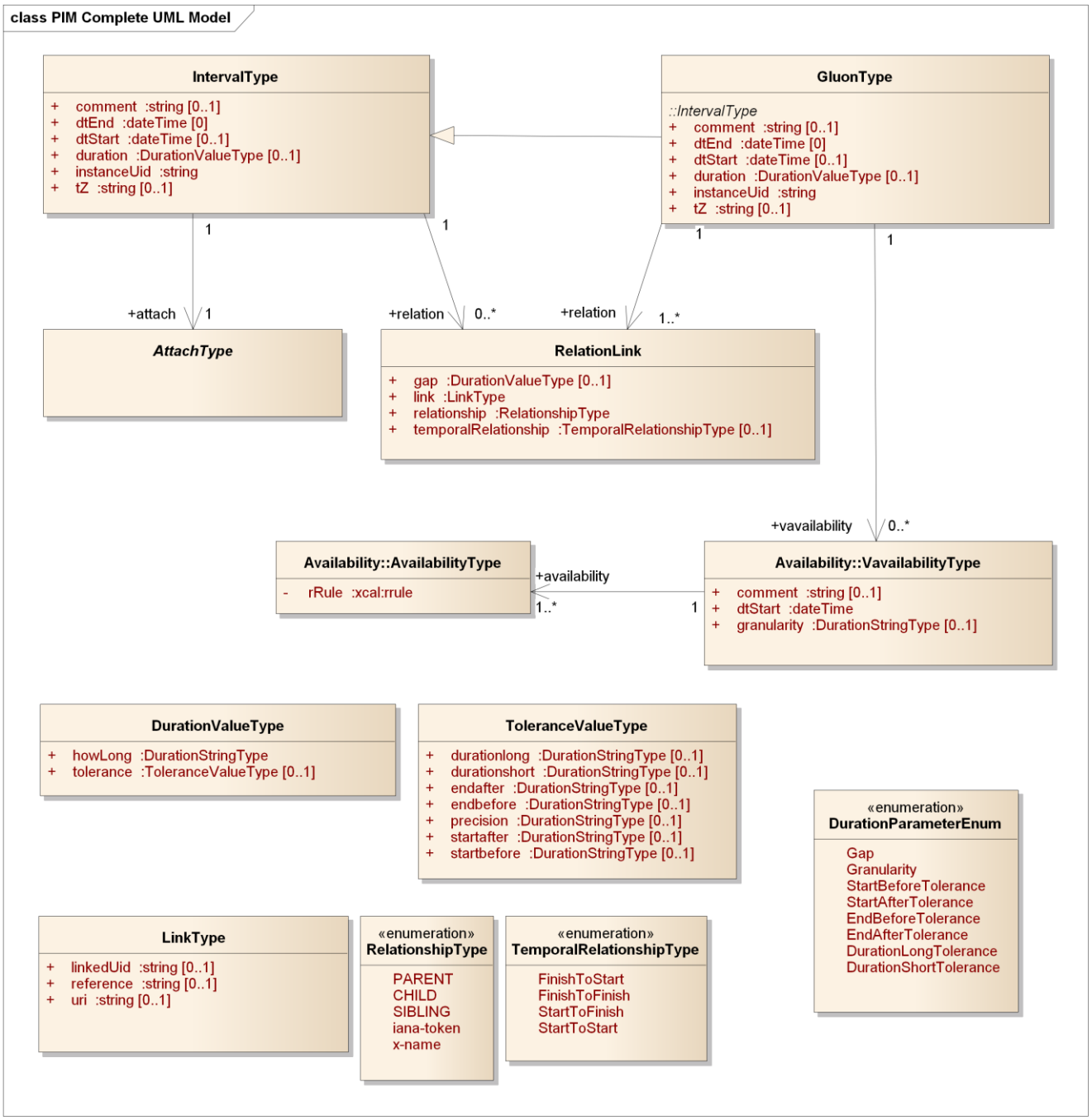

Figure 1 The WS-Calendar PIM UML Model

3.1.2 Discussion

All Associations in PIM are directional; a PSM is NOT required to have identical directionality in associations.

Primitive types express fundamental information related to date, time, and duration, and follow **[RFC5545] [ISO8601]** and are a superset of those expressed in **[iCalendar]**.[4]

Associations in the PIM are directional, but profiles and PSMs derived or derivable from the PIM may have non-directional or variation in the direction of associations to fit their particular platform(s) and purposes.[5]

Attachments are made via the abstract class *AttachType* as described in Section 3.4.

We have used the **[RFC5545] [ISO8601] [iCalendar]** attribute names wherever possible for ease of mapping to that common terminology. In particular, a fully bound Interval is defined by two of

- dtStart—the date & time [dt] for the start of the Interval
- dtEnd—the date & time for the end of the Interval
- duration—the duration (expressed as in **[ISO8601]**) for the Interval

For UML conformance purposes, the three key values for an interval, only two of which are required in fully bound Intervals, are each optional. This permits a conforming instantiation to have zero or more of the three key values; the semantics of Gluons and Intervals in Section 1.9 describes how information for a bound interval is determined. Note also that *GluonType* while a subclass of *IntervalType* has a more restrictive cardinality for *dtEnd* and for *relation*.

3.2 Primitive Types

3.2.1 Introduction

In this section we introduce key concepts and expressions for time including

- DateTime
- Duration
- DurationValueType
- ToleranceValueType

Relationships are described in Section 3.5.

3.2.2 Model Diagram

See IntervalType, DurationValueType, and ToleranceValueType in Figure 1.

The values of the following SHALL be expressed as conformed strings as described in the normative reference **[RFC5545]:**

- DateTime (See Section 3.3.5, Date-Time, in **[RFC5545]**)
- DurationValueType (See Section 3.3.6, Duration, in **[RFC5545]**)

Both DateTime and DurationValueType SHALL permit the full set of **[ISO8601]** date & time and duration conformed strings.

The class

- ToleranceValueType

Is comprised of a set of optional attributes of DurationValueType and is defined in this specification.

3.2.3 Discussion

These concepts are based on **[ISO8601]** and are as expressed in **[iCalendar]** as conformed strings. It is important to note that DurationValueType is *not* the same as that in XML Schema Specification **[XSD]** *Duration*.[6]

---

[4] See discussion in Section 3.2 which includes relationship to **[XSD]** DateTime types.

[5] Note that non-directional associations are a barrier to serializability in messages; hence all PSMs typically would use directional associations unless their purpose is to derive further PSMs.

3.2.4 Relationship to other PIM Components

These concepts are pervasive across the WS-Calendar PIM. The fundamental understanding of time and duration must be consistent and identical to that in **[iCalendar]**. Documentation of any differences in expression MUST be included in the conformance statement for any PSM claiming conformance to this PIM. Moreover, a mapping MUST be provided both directions between the types defined here and those in a PSM claiming conformance.

3.3 Interval

3.3.1 Introduction

The Interval seems to be a simple concept—a bound interval starts at a particular time, runs for a specific duration, and ends at a particular time. This is reflected in Figure 2.

3.3.2 Model Diagram

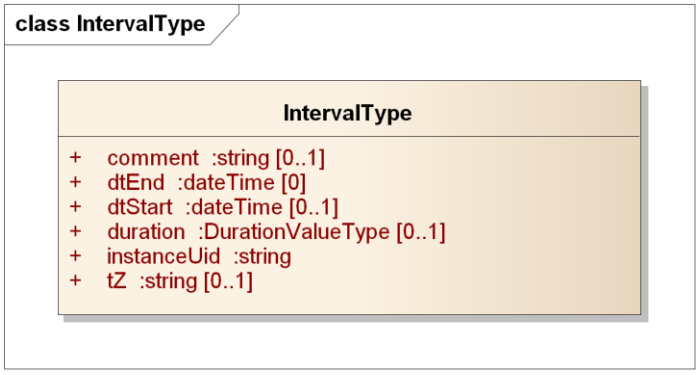

Figure 2 The PIM IntervalType

3.3.3 Discussion

The IntervalType class is the fundamental unit for expressing a time interval; while logically any two or three of the set {dtStart, dtEnd, and duration} can express an interval, there are significant advantages to adopting a single canonical form, particularly one where the semantics are cleanly expressed. Following **[WS-Calendar]**, the canonical PSM that can be managed by standard iCalendar servers, we choose dtStart and duration.

Individual PSMs may use different expressions, but SHOULD recognize in their design that relocation and scheduling of sets of intervals is a very common operation; as we will show later, an entire schedule of Intervals in this WS-Calendar PIM can be scheduled with a single operation, whereas in other representations each dtStart and dtEnd might have to be modified when scheduling.

See also Section 2.5.

3.3.4 Relationship to other PIM Components

The IntervalType class is fundamental to expression of time interval, as are duration and dateTime. Most semantics are with respect to determining values for bound and unbound Intervals.

Payload values are attached to an Interval, as described in the next section.

---

[6] While **[iCalendar]**, **[WS-Calendar],** and this WS-Calendar PIM share the same conforming values, and conform to **[ISO8601]**, **[XSD]** does not include the full specification in **[ISO8601]**. In fact, there are duration strings included in **[XSD]** that are not in **[iCalendar]** and *vice versa*.

3.4 Payload Attachment to an Interval

3.4.1 Introduction

As in **[WS-Calendar]** a payload, which may be comprised of multiple subparts, is attached to an Interval. This differs from

(a) A value containing a description of an Interval (e.g. a measurement that applies to an included Interval)
(b) Associating an interval to a particular measurement (the association is the wrong direction)

The association is directional, and must be completed for use of the Interval instance.

3.4.2 Model Diagram

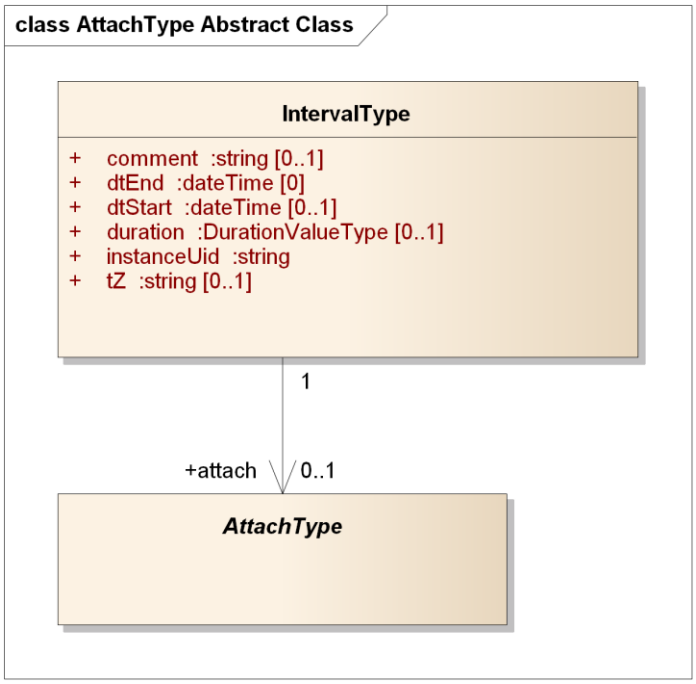

Figure 3 Attaching the Payload to the Interval

3.4.3 Discussion

[WS-Calendar] (line 219) requires that the Attachment Payload and Start Duration must be complete for service performance. In this specification we have defined the cardinality of *attach* to be 0..1 to allow for abstract schedules, including those to which payloads are bound before service performance or concrete use.

3.4.4 Relationship to other PIM Components

The IntervalType is fundamental; application information is attached to instances of the IntervalType by a clear, directional association.

3.5 Relations—Temporal and Other

3.5.1 Introduction

Relationships between IntervalType and its subclasses are accomplished with RelationLinks. The LinkType is defined flexibly to be a UID (as defined in **[xCal]**), a URI **[RFC3986]**, or a reference string.

This supports both distributed schedules and local identifiers that need not be fully qualified as would be a UID or a URI.

There SHALL be at most one of the three attributes of LinkType present in any instance of LinkType.

The TemporalRelationshipType and gap together determine the relationship of the referencing Interval and referenced Interval instances. The gap is the offset; the TemporalRelationshipType express the kind of relationship:

- FinishToStart (the conventional, the referenced interval is after the Finish of the referencing Interval, with an optional gap)
- FinishToFinish (the end of the referencing Interval aligns with the end of the referenced Interval, with an optional gap)
- StartToFinish (the start of the referencing Interval aligns with the end of the referenced Interval, with an optional gap)
- StartToStart (the start of the referencing Interval aligns with the start of the referenced Interval, with an optional gap.

Finally, the RelationshipType expresses whether the relationship is PARENT, CHILD, or SIBLING. The other values in that enumeration are an extension point (x-name) and an IANA-registered xCal token (iana-token) **[IANA]**.

3.5.2 Model Diagram

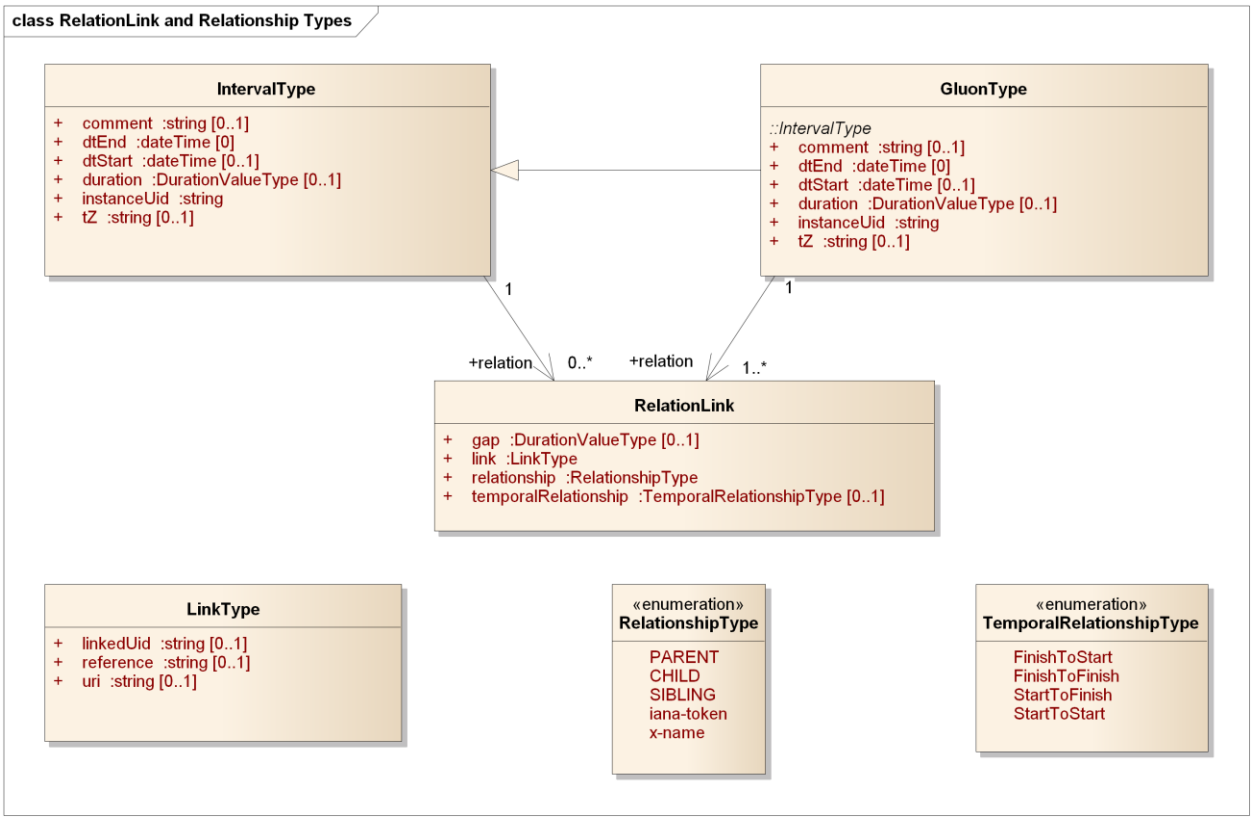

Figure 4 Relation Link and Relationship Types

3.5.3 Discussion

The PIM (and WS-Calendar) supports the common relationships between time intervals, as expressed in schedules, project management tools, and business process definitions such as BPEL and BPMN.

The relationships are expressed using the temporal relationship, the temporal gap between intervals, , and the kind of relationship between Gluons and Intervals expressed as Parent, Child, and Sibling.

316 The set is logically complete, and allows complex structures to be built from primitive relationships,
317 passed in service invocations, and interpreted correctly.

318 3.5.4 Relationship to other PIM Components

319 RelationLinks allow all of the common relationships between time intervals to be expressed with optional
320 offsets (the optional *Gap*), while abstracting the details into the RelationLink class.

321 3.6 Gluons

322 3.6.1 Introduction

323 *GluonType* is a subclass of *IntervalType* with the added requirement that there be at least one
324 *RelationLink*; *IntervalType* may have zero associated *RelationLink*s.

325 The Gluon connects schedules comprised of related Intervals and Gluons, while providing a place for
326 logical information such as the duration of Interval instances so that information may be inherited by
327 referenced Interval instances. The structure permits directed graphs of instances with reuse of
328 components, which may act as reusable sub-schedules.

329 The Gluon may be thought of as a reference into a graph of Intervals or Gluons, allowing differing
330 schedule views depending on the starting point. For example, a room schedule that includes room
331 preparation, meetings, and room cleanup could have a gluon pointing to the preparation Interval for those
332 interested in the preparation starting point and associated actions, and another Gluon pointing to the start
333 of the meetings.

334 A gap is a signed number, so a gap of P-1H for a related interval with TemporalRelationshipType of
335 StartToFinish would mean that the referenced interval starts one hour before the referring interval.

336 These relationships may be used to compose arbitrarily complex graphs of instances of Intervals and
337 Gluons.

3.6.2 Model Diagram

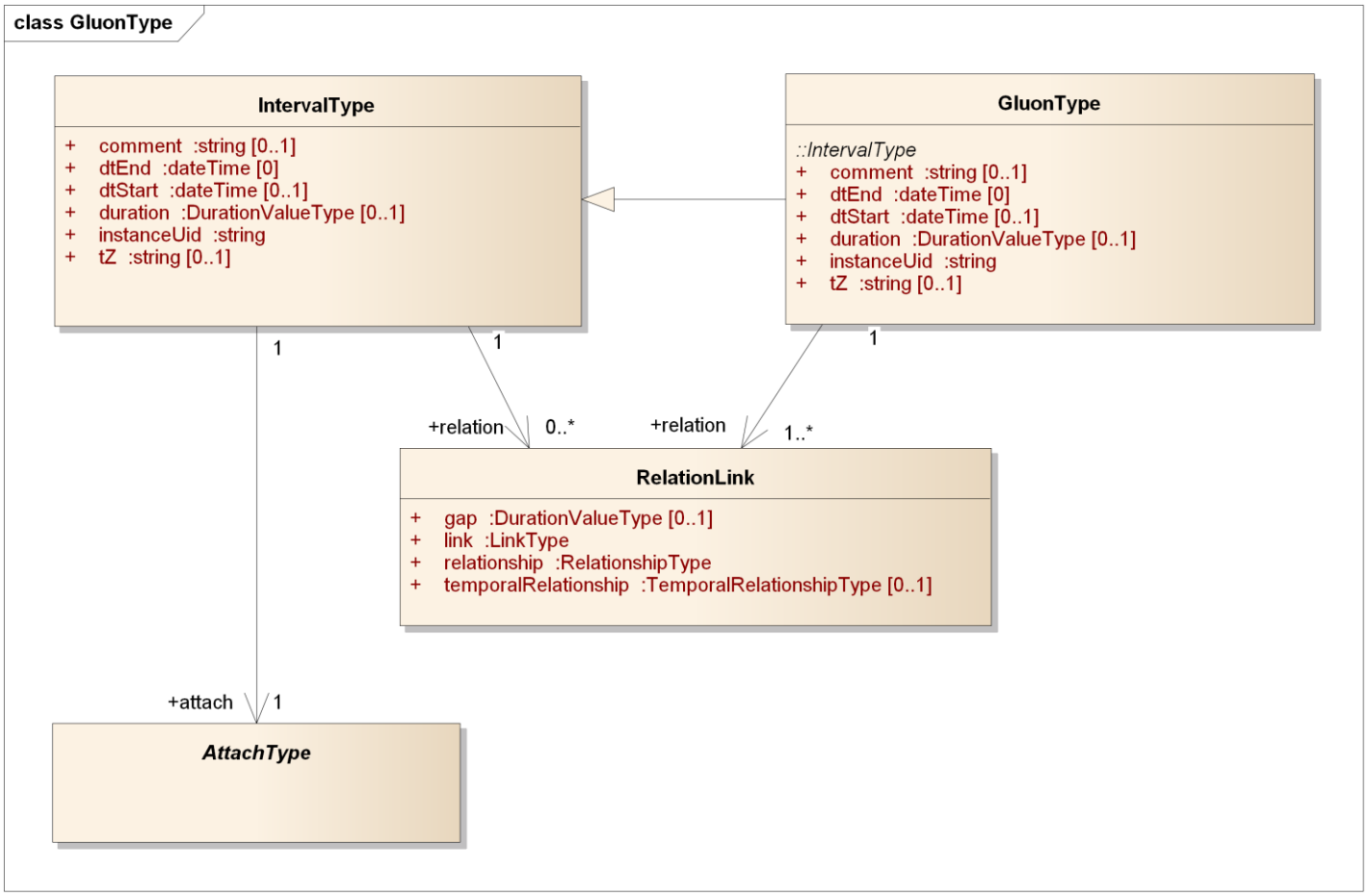

Figure 5 Gluons and their Relationship to Intervals

3.6.3 Discussion

Gluons look and act very much like Intervals. One could think of the Gluon as an optional container for values to “fill in” Interval attributes dynamically and depending on the relationships among the instances.

3.6.4 Relationship to other PIM Components

See the detailed discussion of Semantics in Section 1.9. Gluons contain values that may be inherited or overridden in its children.

3.7 Tolerance and Duration

3.7.1 Introduction

No timing of events, whether descriptive or prescriptive, can be perfectly accurate within the limits of measurement of real systems. The ToleranceValueType is an optional attribute of DurationValueType, allowing full flexibility in the description of permissible or expected variation in duration. See Figure 6.

The characteristics are start early or late, end early or late, or a duration that may be short or long with respect to the nominal value in *howlong*.

3.7.2 Model Diagram

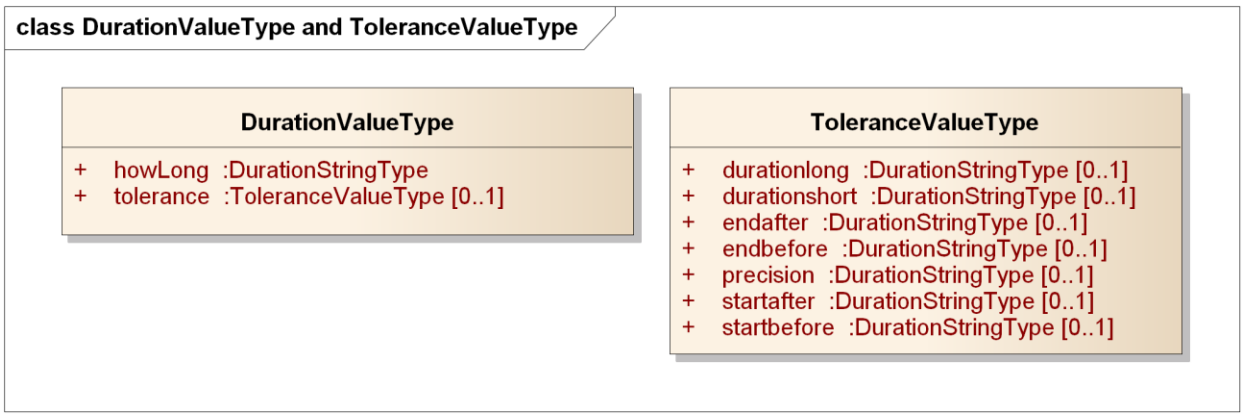

Figure 6 Duration and Tolerance

3.7.3 Discussion

The nature of duration includes a degree and the specification of tolerance with respect to start time, end time, and duration. Tolerances can be expressed in any combination. Relationship to other PIM Components

Any duration, start, or end time as stated (or computed when bound) is subject to tolerance.

"DurationValueType" is the xCal conformed string but is missing some ISO8601 values. This conformed string type is therefore called "DurationStringType" to include the union of xCal and 8601 durations.

3.8 Availability

3.8.1 Introduction

Availability is a means for describing when an actor can be available, or its complement, not available.

This version of the WS-Calendar PIM includes the necessary classes to express Availability as in **[Vavailability]** which is an Internet Draft as of this date.

3.8.2 Model Diagram

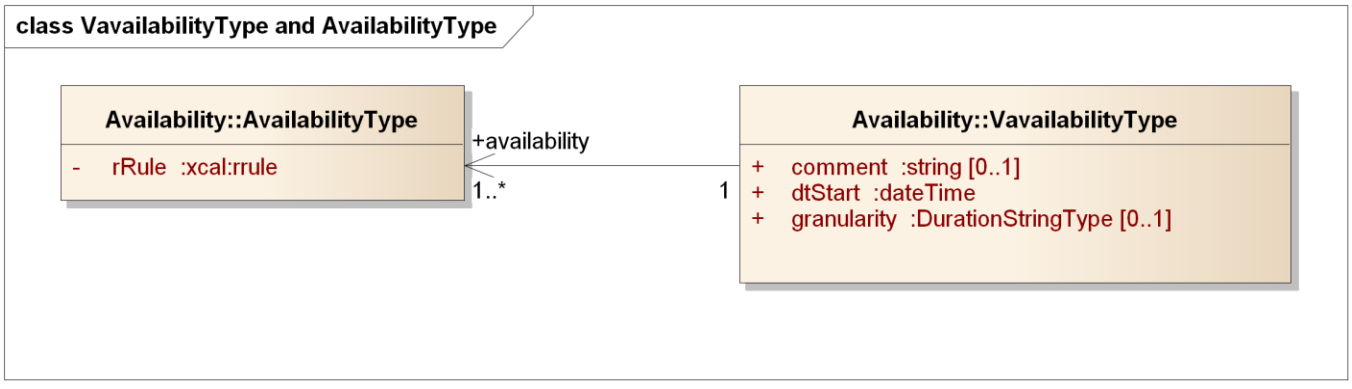

Figure 7 Vavailability and Availability Recurrence Rules

3.8.3 Discussion

The *rrule* is an xCal recurrence rule as defined in section 8.6.5.3 of **[rfc6321]**. The recurrence might be (e.g.) Yearly. In its current form, the expression is in iCalendar syntax, and will need future adaptation to match the abstraction level of this PIM.

3.8.4 Relationship to other PIM Components

Consumes recurrence relationships from Vavailability. Used in consumers of WS-Calendar to express availability for (e.g.) Demand Response events. **[EnergyInteroperation]**. Not used by other parts of the PIM.

4 PIM to WS-Calendar PSM Transformation

MDA instances include a Platform-Independent Model (PIM) which is defined in this specification, and a transformation to a Platform-Dependent Model (PSM). In this section we briefly describe the mapping from this WS-Calendar PIM to **[WS-Calendar]**.

By using the same data types and conformed strings for instance values the transformation is straightforward.

WS-Calendar expresses the information for Intervals, Gluons, and other classes in terms of collections of Parameters, Properties, and Value Types, held in those collections with others that may not reflect the abstractions of WS-Calendar.

Accordingly, the mapping is from the PIM abstractions and classes to the expression in WS-Calendar Parameters, Properties, and Value Types. By the construction of names and types, the relationships of abstract types in the PIM is the same as the relationships of the abstract types in WS-Calendar; the implementation in terms of Parameters, Properties, and Value Types is exactly that in WS-Calendar.

5 Conformance and Rules for WS-Calendar PIM and Referencing Specifications

5.1 Introduction

This Conformance section differs in minor detail from that in **[WS-Calendar]**. The conformance behavior is in general identical to that of WS-Calendar; see Appendix D for details. We do not describe changes in that Appendix that involve the use of the name “WS-Calendar PIM” rather than “WS-Calendar.”

This section specifies conformance related to the information model contained in this specification.

If the implementer and/or implementation claiming conformance is using WS-Calendar PIM as part of a larger business or service communication, they SHALL follow not only the semantic rules herein, but SHALL also conform to the rules for specifying inheritance in referencing standards.

5.2 Relationship to WS-Calendar CS01 [Non-Normative]

This Platform-Independent Model for WS-Calendar shares all of the conformance statements from WS-Calendar CS01 **[WS-Calendar]** subject to

- Renaming of attributes (e.g., *UID* from **[WS-Calendar]** is named *instanceID*)
- Disambiguation of rules (e.g., “Intervals SHALL have a Duration AND (either a dtStart OR a dtEnd)” in 5.3.5.1)
- Simplification (e.g., the section 5.3.5.2)
- Rewording to define conformance to WS-Calendar PIM rather than **[WS-Calendar]**.

5.3 Conformance Rules for WS-Calendar PIM

There are five kinds of conformance that must be addressed for WS-Calendar and specifications that reference WS-Calendar. This PIM references WS-Calendar and requires the same conformance rules.

- Conformance to the ***inheritance rules*** in WS-Calendar, including the direction of inheritance
- ***Specific attributes*** for each type that MUST or MUST NOT be inherited
- ***Conformance rules*** that Referencing Specifications MUST follow
- Description of ***Covarying attributes*** with respect to the Reference Specification
- ***Semantic Conformance*** for the information within the artifacts exchanged

We address each of these in the following sections

5.3.1 Inheritance in WS-Calendar

In this section we define rules that define inheritance including direction.

I1: Proximity Rule Within a given lineage, inheritance is evaluated though each Parent to the Child before what the Child bequeaths is evaluated.

I2: Direction Rule Intervals MAY inherit attributes from the nearest gluon subject to the Proximity Rule and Override Rule, provided those attributes are defined as Inheritable.

I3: Override Rule If and only if there is no value for a given attribute of a Gluon or Interval, that Gluon or Interval SHALL inherit the value for that attribute from its nearest Ancestor in conformance to the Proximity Rule.

I4: Comparison Rule Two Sequences are equivalent if a comparison of the respective Intervals succeeds as if each Sequence were fully Bound and redundant Gluons are removed.

I5: Designated Interval Inheritance [To facilitate composition of Sequences] the Designated Interval in the ultimate Ancestor of a Gluon is the Designated Interval of the composed Sequence. Special conformance rules for Designated Intervals apply only to the Interval linked from the Designator Gluon.

I6: Start Time Inheritance When a start time is specified through inheritance, that start time is inherited only by the Designated Interval; the start time of all other Intervals are computed through the durations and temporal; relationships within the Sequence. The Designated Interval is the Interval whose parent is at the end of the lineage.

5.3.2 Specific Attribute Inheritance

In WS-Calendar and this PIM the following attributes MUST be inherited in conformance to the Rules (same for Gluons and Intervals):

- dtStart
- dtEnd
- Duration
- Designated Interval (Gluon, special upward inheritance rule)
- Tolerance

In WS-Calendar and this PIM the following attributes MUST NOT be inherited

- instanceUid (Gluons and Intervals)
- Temporal Relationships (between Intervals)
- Relationship Links

5.3.3 General Conformance Issues

This specification is general purpose. Standards that claim conformance to this specification may need to restrict the variability inherent in the expressions of Date and Time to improve interoperation within their own interactions. Aspects of Date and Time that may reward attention and conformance statements include:

- **Precision** – Does the conforming specification express time in Hours or in milliseconds. Consider a standard format recommendation.
- **Time Zones and UTC** – Business interactions have a “natural” choice of local, time zone, or UTC based expression of time. Intents may be local, as they tie to the business processes that drive them. Tenders may be Time-zone based, as they are driven by the local business process, but may require future action across changes in time and in time zone. Transaction recording may demand UTC, for complete unambiguity. The specification cannot require one or another, but particular business processes may require appropriate conformance statements.
- **Business Purpose** – Because WS-Calendar is general purpose, it does not distinguish between different exchanges that may have different purposes. For example, a general indication of capability and/or timeliness may be appropriate for a market tender, and an unanchored Sequence may be appropriate. In the same specification, performance execution could require merely the Gluon to Anchor the Interval. If the distinction between Unanchored and Anchored Interval is critical for a set of interactions, the referencing specification SHALL indicate the proper form for a given exchange.

5.3.4 Covarying Elements

Some elements of WS-Calendar and PIM objects may be **covarying**, meaning that they change together. Such elements are treated as a single element for inheritance, they are either inherited together or the child keeps its current values intact. This becomes important if one or more of a covarying set have default values. In that case, if any are present, then inheritance should deem they are all present, albeit some perhaps in their default values.

5.3.5 Conformance of Intervals

5.3.5.1 Intervals

WS-Calendar PIM Intervals SHALL have a Duration.

Intervals MAY have a Start Time.

Intervals SHALL have a Duration AND optionally dtStart. If a non-compliant Interval is received in a service operation with dtEnd, then the dtEnd SHALL be ignored.

Within a Sequence, a maximum of a single Interval MAY have a dtStart or a dtEnd.

5.3.5.2 Other Elements

A Gluon may have a dtStart value.

5.3.6 Conformance of Bound Intervals and Sequences

Actionable services require Bound Intervals as part of a Bound Sequence. Services may include Intervals that are not bound for informational or negotiation purposes. Some of these are modeled and described as constraints in the UML models that have been produced separately.

- Intervals SHALL have values assigned for dtStart and duration, either explicitly or through inheritance
- Intervals SHALL have no value assigned for dtEnd
- Within a Sequence at most the Designated Interval may have dtStart and duration with a value specified or inherited.
- If Sequences are composed to create other Sequences, then the Designated Intervals within the composing Sequence are ignored.
- Any specification claiming conformance to the WS-Calendar PIM MUST satisfy all of the following conditions:
    - Follow the same style of inheritance (per the Rules)
    - Specify attribute inheritability in the specification claiming conformance
    - Specify whether certain sets of elements must be inherited as a group or specify that all elements can be inherited or not on an individual basis

5.4 Conformance Rules for Specifications Claiming Conformance to WS-Calendar PIM

Specifications that claim conformance to the WS-Calendar PIM SHALL specify inheritance rules for use within their specification. These rules SHALL NOT modify the Proximity, Direction, or Override Rules. If the specification includes covariant elements, those elements SHALL be clearly designated in the specification.

Specifications that normatively reference and claim conformance with the WS-Calendar PIM SHALL define the business meaning of zero duration Intervals.

5.5 Security Considerations

The WS-Calendar PIM describes an informational model. Specifications claiming conformance with the WS-Calendar PIM are likely to use the schedule and interval information as but a small part of their overall communications.

Specifications involving communication and messages that claim conformance to this specification should select the communication and select from well-known methods to secure that communication appropriate to the information exchanged, while paying heed to the costs of both communication failure and of inappropriate disclosure. To the extent that iCalendar schedule servers are used, the capabilities of

security of those systems should be considered as well. Those concerns are out of scope for this specification.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Bruce Bartell	Southern California Edison
Chris Bogen	US Department of Defense (DoD)
Edward Cazalet	Individual
Toby Considine	University of North Carolina at Chapel Hill
Robin Cover	OASIS
William Cox	Individual
Sharon Dinges	Trane
Michael Douglass	Rensselaer Polytechnic Institute
Craig Gemmill	Tridium, Inc.
Dave Hardin	EnerNOC
Gale Horst	Electric Power Research Institute (EPRI)
Gershon Janssen	Individual
Ed Koch	Akuacom Inc.
Benoit Lepeuple	LonMark International
Carl Mattocks	Individual
Robert Old	Siemens AG
Joshua Phillips	ISO/RTO Council (IRC)
Jeremy Roberts	LonMark International
David Thewlis	CalConnect

Appendix B. Revision History

Revision	Date	Editor	Changes Made
01	November 15 2012	William Cox	Initial Draft based on contributed models
02	December 20 2012	William Cox	First draft conformance section. Added explanatory text in individual model sections. GluonType is now a subclass of IntervalType, rather than GluonType having an association to IntervalType.
03	January 31, 2012	William Cox	Completed most sections; indicated questions for the TC as “EDITOR’S NOTE”s. Model is the same as for WD02. WD03 contains a quotation with modifications from the WS-Calendar conformance sections.
04	April 10, 2013	William Cox	Update with responses to questions from WD03; minor changes to the model and many clarifications based on meeting discussions. Included differences between the normative semantics and conformance sections and WS-Calendar 1.0 as non-normative Appendices.
05	April 24, 2013	William Cox	Addressed remaining Editor’s Notes from previous Working Drafts. Changed cardinality for attachment from [1..1] to [0..1] in parallel with unbound attributes expressed in UML. Prepared text for public review.

Appendix C. PIM and WS-Calendar Semantics Differences

The following is a non-normative list of changes required to convert the **[WS-Calendar]** Section 1.9 Semantics section to the Semantics section of the PIM.

We have excluded changes to table numbering, page footers, and purely typographic changes such as deletion of extra spaces.

Line numbers are with respect to **[WS-Calendar]** in PDF form.

Line Number	***Change to [WS-Calendar] to PIM***
200-201	Added references to WS-Calendar and PIM tables
202 Table 1-3, Gluon Entry	Changed "…gluon is influences…" to "…gluon influences…" (typographic)
202 Table 1-3, Artifact Entry	Changed first sentence to "An Artifact is the information attached to, and presumably that occurs or is relevant to the time span described by an Interval."
208, Table 1-4, Busy Entry	Changed "Busy often overlays is overlaid by Availability" to "Busy often overlays Availability."

Appendix D. PIM and WS-Calendar Conformance Differences

The following is a non-normative list of changes required to convert the **[WS-Calendar]** Section 4 Conformance and Rules for WS-Calendar and Referencing Specification to Section 5 of the PIM.

We have excluded changes to table numbering, page footers, and purely typographic changes such as deletion of extra spaces. Text was reworded to refer to the PIM rather than WS-Calendar as needed; such changes are not captured here.

Line numbers are with respect to **[WS-Calendar]** in PDF form.

Line Number	***Change to [WS-Calendar] to PIM***
1450-1453 Introduction	Modified Introduction to apply to the WS-Calendar PIM.
1490	Changed “UID (Gluons and Intervals)” to “instanceUid (Gluons and Intervals)”
1491	Added “Relationship Links” to list.
1522-1523	Changed “Duration AND a dtStart OR a dtEnd” to “Duration AND optionally dtStart.” Changed “received with both a dtStart and a dtEnd then the dtEnd SHALL be ignored” to” “received in a service operation with dtEnd then the dtEnd SHALL be ignored.”
1525-1529	Replaced with “A Gluon may have a dtStart value>” Other conditions are excluded by the UML in PIM.
1550	Change “override” to “modify” […the Proximity, Direction, or Override Rules.]