



**Document Title**      **CDISC Study Design Model in XML (SDM-XML)**

**Release**                **Version 1.0**

**Copyright**              ©2008-2011, Clinical Data Interchange Standards Consortium, Inc.  
All rights reserved.

## Table of Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>4</b>
1.1	SPECIFICATION GOALS	4
1.2	DOCUMENT CONVENTIONS	5
1.3	SPECIFICATION SCOPE: DESIGN, NOT EXECUTION	5
<b>2</b>	<b>CONFORMITY AND GENERAL ISSUES</b>	<b>5</b>
2.1	FILE CONFORMITY	6
2.2	VENDOR EXTENSIONS	6
2.3	INCLUSION IN ODM DOCUMENTS	6
2.4	USE OF THE ORDERNUMBER ATTRIBUTE	7
2.5	DEFS AND REFS	7
2.6	PARTIAL DEFINITIONS	7
<b>3</b>	<b>PROTOCOL SUMMARY AND STUDY INCLUSION/EXCLUSION CRITERIA</b>	<b>7</b>
3.1.1	<i>Summary</i>	7
3.1.2	<i>InclusionExclusionCriteria</i>	8
<b>4</b>	<b>STRUCTURAL ELEMENTS</b>	<b>8</b>
4.1	BACKGROUND CONCEPTS: EPOCHS, ARMS, CELLS, SEGMENTS	8
4.2	STRUCTURAL ELEMENTS	9
4.2.1	<i>Epoch</i>	9
4.2.2	<i>Arm</i>	10
4.2.3	<i>CellDef</i>	10
4.2.4	<i>SegmentDef</i>	12
4.3	ACTIVITIES	12
4.3.1	<i>Activities for Form Completion</i>	13
4.3.2	<i>Other Purposes for an Activity</i>	13
4.4	STUDY EVENTS	13
<b>5</b>	<b>WORKFLOW</b>	<b>15</b>
5.1	STUDY START, STUDYFINISH, PATHCANFINISH	15
5.1.1	<i>StudyStart</i>	15
5.1.2	<i>StudyFinish</i>	15
5.1.3	<i>PathCanFinish</i>	15
5.2	ENTRY AND EXIT CRITERIA	16
5.3	WORKFLOW PATHS AND BRANCHING	17
5.3.1	<i>Transition</i>	17
5.3.2	<i>Switch</i>	18
5.4	SPECIFYING CONTINGENCIES FOR UNPLANNED EVENTS	20
5.4.1	<i>Trigger</i>	20
5.5	PROCESSING TRANSITIONS BETWEEN ACTIVITIES	21
5.6	WORKFLOW TRANSITIONS BETWEEN VISITS, CELLS, EPOCHS AND SEGMENTS	21
<b>6</b>	<b>TIMING</b>	<b>21</b>
6.1	TIMING CONCEPTS	22
6.1.1	TIMING WINDOWS	22
6.1.2	TIMEPOINT GRANULARITY	22
6.1.3	TIMING TYPES	22

6.1.4	SUBSEQUENT SCHEDULING BASIS .....	23
6.2	ABSOLUTE TIMING CONSTRAINTS.....	23
6.3	RELATIVE TIMING CONSTRAINTS.....	23
6.4	TRANSITION TIMING CONSTRAINTS .....	24
6.5	ACTIVITY DURATIONS .....	25
6.6	EVALUATION OF TIMING CONSTRAINTS .....	26
6.7	TIMING CONSTRAINTS ON VISITS, CELLS, EPOCHS AND SEGMENTS.....	27
<b>7</b>	<b>ELEMENT REFERENCE .....</b>	<b>27</b>
7.1	ODM V1.3.1 ELEMENTS USED BY STUDY DESIGN .....	27
7.2	STUDY DESIGN-SPECIFIC ELEMENTS .....	27
<b>8</b>	<b>EXAMPLES .....</b>	<b>27</b>
<b>9</b>	<b>XML SCHEMA FOR SDM-XML .....</b>	<b>27</b>
<b>10</b>	<b>ACKNOWLEDGMENTS .....</b>	<b>28</b>

# 1 Introduction

A study protocol is a plan for executing a study, and part of that plan is the design of the study. CDISC has used the term “study design” to include:

- The experimental design of the study. Experimental designs can be classified using terms such as “parallel”, “cross-over” or “factorial”, but such a term is not enough to describe the experimental design for a particular study. A complete description specifies, for each of the arms to which a subject may be assigned, the sequence of “treatment strategies” to which the subject will be exposed, where “treatment strategy” is to be interpreted broadly to include preparation for treatment, non-treatment, placebo treatment, withdrawal from treatment, and post-treatment, as well as active treatments. In study protocol documents, the experimental design of the study is often represented in a diagram called a study schema.
- The schedule of activities for the study. This includes planned visits or other times during which treatments and assessments are to be done, the planned treatments, assessments and other activities to be performed, and the plans for which activities are to be done at which times. The plan for a particular activity to be done at a particular time may depend on various conditions (e.g., whether the subject is female, whether the subject has experienced a certain adverse event, or had a certain abnormal lab result). Indeed, whole visits may depend on certain conditions (e.g., early withdrawal visits, delays to visits due to delays in treatments). In study protocol documents, the schedule of activities is often represented in tabular form, although conditions and exceptions often mean that the table must be supplemented with multiple footnotes.
- The eligibility criteria for the study.
- Certain summary information about a study, such as its title, phase, objectives, sex, age, and other characteristics of eligible subjects, number of subjects, and planned duration.

Several CDISC standards represent aspects of study design:

- The Operational Data Model (ODM) represents metadata for the data collected in a study, and so is closely aligned with the schedule of activities, although it does not describe the planned timing of the groups of data called “study events”, which are implicitly assumed generally to represent the data collected at a certain visit.
- The Study Data Tabulation Model (SDTM) includes Trial Design datasets which represent summary information, eligibility criteria, the experimental design, and planned visits. These datasets do not include planned activities or times other than those designated as “visits” and so do not describe the full schedule of activities.
- The Protocol Representation Model (PRM) is a conceptual model, documented in the Unified Modeling Language (UML) of the concepts of a clinical study protocol. Version 1 includes all of study design, as defined above.

This document describes a means of specifying a study design as an extension to core ODM.

## 1.1 Specification Goals

In providing an ODM-based XML representation of study design, this specification aims to:

- Build upon the foundation laid by the ODM in representing aspects of a study in XML.
- Leverage the ODM data types wherever appropriate.
- Use existing ODM constructs where possible.
- Enable the design-time use of reusable building blocks specific to study design.
- Provide enough information on which an *execution runtime* could operate to follow individual study participants.

## 1.2 Document Conventions

This document describes study design XML elements, their attributes, and their relationships. In addition to plain text, figures and examples also are given.

Examples of the raw XML representing study design also are given throughout this document. Many of these examples are not complete XML documents, but rather XML fragments that would be placed at the appropriate point in a containing ODM document. In these examples, Study Design Model elements are given the prefix "sdm:", while ODM elements are shown as belonging to the default namespace (no prefix).

The terms "element" and "attribute" are used throughout this document. These are used to reference the XML constructs of the same name, or alternatively, to reference objects and properties of those objects, respectively. In particular, the term "element" as used in this document is not to be confused with the corresponding term's usage in describing particular CDISC SDTM datasets.

## 1.3 Specification Scope: Design, Not Execution

The study design model is just that – a model of a study's *design*. It is not a record of an individual study participant's route through the study. An execution runtime would reference the design elements to decide how a study participant should progress through the study and record this path accordingly.

The study design model separates distinct aspects of study design into *sub-components*:

- **Structure:** Arms, Cells, Segments, Activities, etc.
- **Workflow:** How a participant should progress through a study – decision points, branches, etc.
- **Timing:** When the activities should happen, usually relative to other structural elements of the study.

Structural constructs can reference only existing ODM constructs. Workflow constructs can reference only existing ODM constructs and Structural constructs. Timing constructs can reference either structural, workflow, or existing ODM constructs.

Study execution, on the other hand, would need to capture at least the following:

- The path that each individual study participant took through the study design.
- The activities that were performed on that path.
- The planned times for when the activities should have happened for that study participant.
- The actual times that the activities were performed and decision points reached.
- References to the design elements for information that is common to all study participants.

Study execution is outside the scope of this specification. However, the study design model was developed with the intent of ensuring that an execution runtime would have sufficient information to operate over a study design. (The consumption of a study design representation by a study-execution "runtime" sometimes is referred to as 'protocol insertion', or as the retrieval of a protocol for execution).

## 2 Conformity and General Issues

This section supplements the corresponding section, "General Issues", of the ODM v1.3.1 specification. All conformity requirements described in the ODM v1.3.1 specification are applicable to any ODM v1.3.1 file that contains Study Design Model XML.

## 2.1 File Conformity

The namespace URI for version 1.0 of the XML Study Design Model is:  
**http://www.cdisc.org/ns/studydesign/v1.0**

Throughout this document, all ODM elements are referenced as being in the default namespace (they have no namespace prefix). Elements in the Study Design Model namespace are assigned the prefix "sdm". While any distinct namespace prefixes may be used in practice, for readability and ease of comparison we recommend using the namespace prefixing shown in this document.

## 2.2 Vendor Extensions

The XML Study Design Model schema permits vendor extensions, as defined in the ODM 1.3.1 specification, to the elements defined in this specification.

## 2.3 Inclusion in ODM Documents

This example shows the XML that would comprise the minimal "shell" of any ODM 1.3.1 document that contains Study Design markup. Note that the Protocol element within a MetaDataVersion element acts as the container for the Study Design definitions.

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:sdm="http://www.cdisc.org/ns/studydesign/v1.0"
  Description="CDISC ODM v1.3.1 with SDM Definitions"
  FileType="Transactional" Granularity="All"
  FileOID="SDM_Example_00" CreationDateTime="2009-03-20T11:07:23-05:00">
  <Study OID="SAMPLE_STUDY">
    <GlobalVariables>
      <StudyName>CDISC Study Design Prototype</StudyName>
      <StudyDescription>A sample study</StudyDescription>
      <ProtocolName>SDM (Prototype)</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion Description="Version 1.0.0"
      Name="Version 1.0.0" OID="v1.0.0">
      <Protocol>
        <!-- ***** -->
        <!-- === Study Design: Structural Elements/Workflow/Timing === -->
        <!-- ***** -->
      </Protocol>
      <!-- ***** -->
      <!-- ===== CORE/SHARED ODM DEFINITIONS ===== -->
      <!-- ***** -->
    </MetaDataVersion>
  </Study>
</ODM>
```

**Example 2.3:** Overall structure of an ODM document containing study design markup.

Other references between ODM elements and Study Design elements may exist, including references by Study Design elements to ODM ConditionDefs, references by ODM StudyEventDefs to Study Design ActivityDefs, and references from Study Design ActivityDefs to ODM FormDefs. Such relationships across the ODM and Study Design namespaces are described in this document.

## 2.4 Use of the OrderNumber Attribute

In several locations, the XML Study Design specification uses an optional OrderNumber attribute to permit the definition of relative ordering of elements within a container. In the context of SDM, if the OrderNumber attribute is specified on an element, all elements of the same type (element name) within the same parent element must also specify an OrderNumber attribute. It is an error to mix elements with OrderNumber and without OrderNumber within a given parent element. If no OrderNumber attribute is specified in cases where order is important, the ordering of elements is to be derived from the XML document order.

## 2.5 Defs and Refs

This document sometimes references elements as *Defs* and *Refs*. In SDM-XML, a 'Def' is the declaration of an object instance. A 'Ref' is a reference to that object from some other entity; SDM-XML Refs do not imply instantiation.

For example, in this specification segments are declared using elements named 'SegmentDef'. However, to indicate that a cell includes a particular segment instance, a SegmentRef will be employed to reference the appropriate SegmentDef object.

## 2.6 Partial Definitions

As with ODM, SDM-XML permits the definition of partial/incomplete information regarding a clinical study. For example, the SDM-XML Schema permits the definition of a Cell that contains zero segments. Such a definition, while incomplete in the context of what generally is required for a study definition to be considered complete, may yet be useful in various applications, including incremental development of study designs.

# 3 Protocol Summary and Study Inclusion/Exclusion Criteria

Most of the study design elements within the ODM 1.3.1 Protocol elements exist within one of three categories: Structure, Workflow, and Timing. However, a small number of elements exist as annotations to the ODM Protocol element itself and are considered not to be exclusively structural, workflow, or timing in nature. In the study design model, these elements are the Summary and protocol-level InclusionExclusionCriteria.

### 3.1.1 Summary

The **Summary** element provides the ability to define a set of parameters to the study design. Each **Parameter** subelement is assigned an OID unique to the protocol. In addition, each parameter includes a term and optional short name, and comprises one or more corresponding values.

```
<sdm:Summary>
  <sdm:Parameter OID="PAR.AGESPAN" Term="Age Span" ShortName="AGESPAN">
    <sdm:Value>Adult (18-65)</sdm:Value>
    <sdm:Value>Elderly (&gt;65)</sdm:Value>
  </sdm:Parameter>
</sdm:Summary>
```

**Example 3.1.1:** The study summary element.

### 3.1.2 InclusionExclusionCriteria

The **InclusionExclusionCriteria** element can contain two lists of **Criterion** elements, represented by the two elements **InclusionCriteria** and **ExclusionCriteria**. Together, these criteria determine the eligibility of a subject for the study. The actual condition to be evaluated is contained in an ODM ConditionDef, which is referenced by each Criterion's ConditionOID attribute.

The point in time at which these criteria are evaluated relative to the structural elements of the study design is dependent on the processing runtime and workflow configuration. Individual criteria may be evaluated by referencing the relevant conditions in transitions. This could, for example, force the subject's execution path into a study-finish state.

The **InclusionExclusionCriteria** element also may contain an optional description.

```
<sdm:InclusionExclusionCriteria>
  <Description>
    <TranslatedText xml:lang="en">Include subjects which ...</TranslatedText>
  </Description>
  <sdm:InclusionCriteria>
    <sdm:Criterion OID="CRIT00" ConditionOID="AGECOND00" Name="Age Inclusion"/>
  </sdm:InclusionCriteria>
  <sdm:ExclusionCriteria>
    <sdm:Criterion OID="CRIT01" ConditionOID="AGECOND01" Name="Age Exclusion"/>
    <sdm:Criterion OID="CRIT02" ConditionOID="PREGNANCYCOND"
      Name="Pregnancy Exclusion"/>
  </sdm:ExclusionCriteria>
</sdm:InclusionExclusionCriteria>
```

**Example 3.1.2:** Study inclusion and exclusion criteria.

## 4 Structural Elements

Structural elements are comprised of the “building blocks” of a study design: objects such as Epochs, Cells, Arms and Segments, as well as Activities. These are the objects that can act as nodes in a study design workflow, or as objects between which timing constraints may be applied.

### 4.1 Background Concepts: Epochs, Arms, Cells, Segments

In this section, we review the major structural concepts in existing study design practice.

The planned period of subject participation in a study is divided into **epochs**. Each epoch is a period of time which serves a purpose in the study as a whole. Typically, the purpose of an epoch is to expose a subject to treatment (eg: Treatment), prepare for treatment (eg: Screening, Washout) or to gather data on a subject after treatment has ended (eg: Followup).

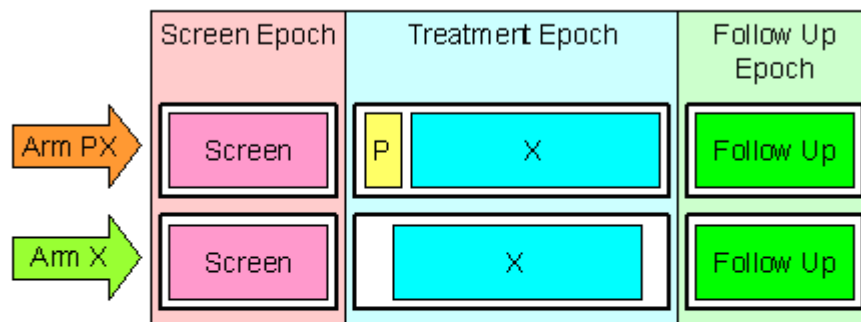
A study **arm** represents one planned path through the study. The path is composed of a study cell for each epoch in the study.

A study **cell** is the part of study design that describes what happens in a particular epoch for a particular arm. The cell describes how the purpose of its epoch is fulfilled for each arm.

**Segments** are often seen as the basic building blocks of study design. A segment usually specifies a combination of planned observations and interventions, which may or may not involve treatment, during a period of time.

The diagram below illustrates the relationships between these concepts.

A two-arm trial comparing Treatment X with and without Pre-treatment P



Columns shown with large rectangles "in back" are epochs.

Rows marked by arrows are arms.



Rectangles with heavy outlines are trial cells.



Rectangles within the trial cells are segments.

In this specification, we additionally and formally define the concept of an **activity**, one or more of which may be included within each segment.

An ODM StudyEvent, which often represents the concept of a "visit," also is a structural element of study design. It too can be thought of as being contained within a study segment. Although the relationship between activities and "visits" cannot be enforced in a rigid hierarchy, the normal case is that, at the time a study is planned, activities can often be thought of as taking place in the context of a StudyEvent.

## 4.2 Structural Elements

This section describes the elements that define the overall structure of a study.

### 4.2.1 Epoch

A set of **Epoch** elements represent the study's Epochs. A protocol may include definitions of one or more epochs. Since Epochs are sequential blocks of time for a study, some ordering of the epochs relative to one another is also required. This ordering is derived from the XML Document order or an optional OrderNumber attribute as described in section 2.4. An Epoch contains an optional description.

An example of the XML for Epoch definitions:

```

<sdm:Epoch OID="SCREPOCH" Name="Screening Epoch" OrderNumber="1">
  <Description>
    <TranslatedText xml:lang="en">Screening Epoch</TranslatedText>
  </Description>
</sdm:Epoch>
<sdm:Epoch OID="TREPOCH" Name="Treatment Epoch" OrderNumber="2">
  <Description>
    <TranslatedText xml:lang="en">Treatment Epoch</TranslatedText>
  </Description>
</sdm:Epoch>

```

**Example 4.2.1:** Epoch definitions.

## 4.2.2 Arm

An **Arm** element provides the declaration of a study arm. A Protocol element may include definitions of one or more arms. Arms do not have any ordering relative to one another.

```

<sdm:Arm OID="PLACEBO_ARM" Name="Placebo Arm" >
  <Description>
    <TranslatedText xml:lang="en">Placebo arm</TranslatedText>
  </Description>
</sdm:Arm>
<sdm:Arm OID="LOWDOSE_ARM" Name="Low Dose Arm" >
  <Description>
    <TranslatedText xml:lang="en">
      Low-dose arm (54 mg) xanomeline)
    </TranslatedText>
  </Description>
</sdm:Arm>
<sdm:Arm OID="HIGHDOSE_ARM" Name="High Dose Arm" >
  <Description>
    <TranslatedText xml:lang="en">High-dose arm (81 mg)</TranslatedText>
  </Description>
</sdm:Arm>

```

**Example 4.2.2:** Arm definitions.

## 4.2.3 CellDef

A **CellDef** provides for the declaration of a study cell, which is a common study-design construct used for representing the intersection of an epoch with an arm. While some cells have an ordering relative to one another, any such ordering can be derived from the ordering of the epochs to which the cells are associated.

Each CellDef must reference exactly one Epoch to which it belongs, and reference zero or more Arms. A cell may belong to zero Arms when it represents a part of the study with only one path (screening, for example, which is the same for all participants). A Cell may be part of exactly one Arm in the case of an unblinded study. However, in the case of a blinded study, no foreknowledge of a single Arm can exist. Therefore, this model permits a cell in a blinded study to reference the one or more possible arms on which activities within the cell may take place. Only at runtime, during randomization/assignment, will the study execution mechanism determine to which of the possible arms a subject has actually been assigned.

A CellDef may reference zero or more SegmentDef elements via SegmentRef elements. Each such reference indicates a segment of which the cell is composed. Any given SegmentDef should never be

referenced more than once from the set of all CellDefs for a study. This constraint is necessary in order to ensure that a valid workflow can be defined using workflow components of SDM.

The order of SegmentRefs within a cell is derived from the XML Document order or an optional OrderNumber attribute as described in section 2.4. Any such ordering exists only for potential use in presenting a list of segments to a user, and does not imply anything about event scheduling, time ordering, or data correctness. In SDM-XML, scheduling and time ordering are determined by definitions given in SDM-XML Workflow and Timing sections.

Examples for both unblind and blind scenarios are given below. First is an example of CellDef XML for a study with no blinding. In this example, three cells are defined. The first is a screening cell, and has no arm assignment. The second and third both reference the same Epoch, but occur within different Arms.

```
<sdm:CellDef OID="SCREENCELL" Name="Screening Cell" EpochOID="SCREPOCH">
  <Description>
    <TranslatedText xml:lang="en">Screening cell</TranslatedText>
  </Description>
  <sdm:SegmentRef SegmentOID="SCREENSEG" OrderNumber="1" />
</sdm:CellDef>

<sdm:CellDef OID="DRUGCELL" Name="Drug Treatment Cell" EpochOID="TREPOCH">
  <Description>
    <TranslatedText xml:lang="en">Drug treatment cell</TranslatedText>
  </Description>
  <sdm:ArmAssociation Type="Unblinded">
    <sdm:ArmRef ArmOID="DRUG_ARM"/>
  </sdm:ArmAssociation>
  <sdm:SegmentRef SegmentOID="DRUGSEG" OrderNumber="2" />
</sdm:CellDef>

<sdm:CellDef OID="OPERCELL" Name="Op Treatment Cell" EpochOID="TREPOCH">
  <Description>
    <TranslatedText xml:lang="en">Operative procedure cell</TranslatedText>
  </Description>
  <sdm:ArmAssociation Type="Unblinded">
    <sdm:ArmRef ArmOID="OPER_ARM"/>
  </sdm:ArmAssociation>
  <sdm:SegmentRef SegmentOID="OPERSEG" OrderNumber="3" />
</sdm:CellDef>
```

**Example 4.2.3.a:** Cell definitions, without blinding.

The next example shows CellDef XML for a blinded study. In this example, only two CellDefs are given – a screening cell and a treatment cell. The screening cell’s definition remains the same as in the unblinded example. However, since this is a blinded study, the treatment cell cannot be assigned to a particular arm at design-time. Rather, a list of possible arm assignments is given within the cell’s ArmAssociation element.

```

<sdm:CellDef OID="SCREENCELL" Name="Screening Cell" EpochOID="SCREPOCH">
  <Description>
    <TranslatedText xml:lang="en">Screening cell</TranslatedText>
  </Description>
  <sdm:SegmentRef SegmentOID="SCREENSEG" OrderNumber="1" />
</sdm:CellDef>

<sdm:CellDef OID="TRECELL" Name="Treatment Cell" EpochOID="TREPOCH">
  <Description>
    <TranslatedText xml:lang="en">Treatment cell</TranslatedText>
  </Description>
  <sdm:ArmAssociation Type="Blinded">
    <sdm:ArmRef ArmOID="PLACEBO_ARM"/>
    <sdm:ArmRef ArmOID="LOWDOSE_ARM"/>
    <sdm:ArmRef ArmOID="HIGHDOSE_ARM"/>
  </sdm:ArmAssociation>
  <sdm:SegmentRef SegmentOID="TRESEG" OrderNumber="2" />
</sdm:CellDef>

```

**Example 4.2.3.b:** Cell definitions, with blinding.

## 4.2.4 SegmentDef

A **SegmentDef** represents a set of activities. Each segment must be referenced by a cell via a SegmentRef within a CellDef.

A SegmentDef may reference zero or more ActivityDef elements. Each such reference indicates an activity which is to be undertaken as part of that segment. Any given ActivityDef should never be referenced more than once from the set of all SegmentDefs for a study. This constraint is necessary in order to ensure that a valid workflow can be defined using workflow components of SDM.

The order of ActivityRefs within each SegmentDef is derived from the XML Document order, or an optional OrderNumber attribute, as described in section 2.4. Any such ordering exists only for potential use in presenting a list of activities to a user, and does not imply anything about event scheduling, time ordering, or data correctness. In SDM-XML, scheduling and time ordering are determined by definitions given in SDM-XML Workflow and Timing sections.

An example of the XML for segment definitions:

```

<sdm:SegmentDef OID="SCREENSEG" Name="Screening Segment">
  <Description>
    <TranslatedText xml:lang="en">Screening segment/period</TranslatedText>
  </Description>
  <sdm:ActivityRef ActivityOID="ECGFORM"/>
  <sdm:ActivityRef ActivityOID="VSFORM"/>
</sdm:SegmentDef>

```

**Example 4.2.4:** Segment definition.

## 4.3 Activities

An **Activity** represents a point in a study at which a specific action is to be taken.

### 4.3.1 Activities for Form Completion

A common purpose for an activity in a study is for completion of a form. A study design can specify this purpose for an activity by including one or more ODM FormRef elements, referencing the definition of the forms that must be filled in when the activity is undertaken; this is done by specifying a reference to the forms as a child of the given Activity's definition element, as shown in the example below.

```
<sdm:ActivityDef OID="ACTDEF_VITALSIGNS_01" Name="Collect Vital Signs">
  <Description>
    <TranslatedText xml:lang="en">Collection of Vital Signs</TranslatedText>
  </Description>
  <FormRef FormOID="VITFORM" Mandatory="Yes"/>
  <FormRef FormOID="MEDFORM" Mandatory="Yes"/>
</sdm:ActivityDef>
```

**Example 4.3.1:** Definition of an activity, with form completion.

### 4.3.2 Other Purposes for an Activity

An Activity also allows designers to include timing and follow transition rules that are not associated directly with a form (this type of activity is sometimes referred to as a 'Milestone'). Such activities would be referenced from elements in the workflow or timing sections of SDM in order to build up the appropriate runtime characteristics for the study.

```
<sdm:ActivityDef OID="ACTDEF_SCHEDULING" Name="Schedule Assessment"/>
```

**Example 4.3.2:** Definition of an activity, without form completion.

## 4.4 Study Events

ODM 1.3.1 itself provides for the definition of a study event. The Study Design Model extends the ODM StudyEventDef, allowing it to reference any number of activity definitions.

StudyEvents can be associated with no type of structural element other than ActivityDefs. This permits the definition of a visit simply as a group of activities that are convenient to perform together.

There is no requirement that an Activity be contained within a StudyEvent. In fact, it occasionally may be desirable to define an Activity that will occur outside of a StudyEvent.

Activities can be assigned a default order within a study event, but this ordering may be overridden at runtime by the actual path specified in the study design's workflow definition.

```

<!-- In MetaDataVersion -->
<StudyEventDef Name="Treatment Visit 12" OID="TRT_VISIT_12"
    Repeating="No" Type="Scheduled">
  <FormRef FormOID="VITFORM" Mandatory="Yes"/>
  <FormRef FormOID="CMFORM" Mandatory="Yes"/>
  <sdm:ActivityRef ActivityOID="ACT_VS" OrderNumber="1"/>
  <sdm:ActivityRef ActivityOID="ACT_ADVASSESS" OrderNumber="2"/>
  <sdm:ActivityRef ActivityOID="ACT_DISPENSE_MED" OrderNumber="3"/>
</StudyEventDef>

<!-- In sdm:Structure -->
<sdm:ActivityDef OID="ACT_VS" Name="Collect Vital Signs">
  <FormRef FormOID="VITFORM" Mandatory="Yes"/>
  <FormRef FormOID="CMFORM" Mandatory="Yes"/>
</sdm:ActivityDef>

```

**Example 4.4:** Definition of activity references for a study event.

Note that an Activity referenced from within a study event may be a data-collection activity that references forms. Therefore, the StudyEventDef (via the ActivityDef referenced via ActivityRef) will have an indirect reference to forms to be filled in during the study event. Such forms must also be directly referenced from the StudyEventDef using an ODM FormRef element in order to ensure that ODM metadata matches the resulting Clinical Data. This is required, since metadata and ClinicalData must match even when all SDM extensions are removed from the Study Design XML in order for the containing markup to be considered valid ODM.

## 5 Workflow

Study workflows are defined using a set of constructs that make it possible for a study designer to specify possible study participant paths through a study.

Workflow is specified in a section of XML distinct from that of the structural elements. However, workflow objects commonly reference objects defined in the Structure section of the document. This separation of concerns allows the potential for different workflow representations to be applied to the same set of structural elements.

### 5.1 Study Start, StudyFinish, PathCanFinish

The SDM-XML workflow section supplies constructs for marking an activity as having a workflow-specific role.

#### 5.1.1 StudyStart

A protocol should contain exactly one "Study Start" activity to identify a single entry point into the study for a given participant. The presence of an activity marked as the study start is useful as an anchor for timings relative to the start of a study.

The **StudyStart** element has an optional ODM Description child element and must contain exactly one ActivityRef child element. For example:

```
<sdm:StudyStart>
  <sdm:ActivityRef ActivityOID="ACTDEF_STUDYSTART"/>
</sdm:StudyStart>
```

**Example 5.1.1:** Definition of a study-start activity.

The optional OrderNumber attribute on sdm:ActivityRef has no meaning here, and, when present, should be ignored by a receiving system.

#### 5.1.2 StudyFinish

A protocol should contain exactly one "Study Finish" activity, which explicitly identifies the point at which a study participant's path through the study has finished.

The **StudyFinish** element has an optional ODM Description child element and must contain exactly one ActivityRef child element. For example:

```
<sdm:StudyFinish>
  <sdm:ActivityRef ActivityOID="ACTDEF_STUDYFINISH"/>
</sdm:StudyFinish>
```

**Example 5.1.2:** Definition of a study-finish activity.

The optional OrderNumber attribute on sdm:ActivityRef has no meaning here, and, when present, should be ignored by a receiving system.

#### 5.1.3 PathCanFinish

The **PathCanFinish** element allows you to specify activities for which it is not necessarily an error that the workflow does not provide a transition from that activity to another. When, during execution in a runtime, an activity marked as "PathCanFinish" is reached in traversing a given subject's path,

the runtime is permitted to treat this as a normal condition. Possible actions by the runtime include the raising of an alert to system users, and/or requiring manual intervention to determine the next activity to be undertaken. This can be especially useful when handling paths that were begun via the triggering of an unplanned event, for which the "return path" cannot be fully anticipated at study-design time.

The PathCanFinish element has an optional ODM Description child element and may contain zero or more ActivityRef child elements. For example:

```
<sdm:PathCanFinish>
  <sdm:ActivityRef ActivityOID="ACT.FeverTriggerResolved"/>
  <sdm:ActivityRef ActivityOID="ACT.BPTriggerResolved"/>
</sdm:PathCanFinish>
```

**Example 5.1.3:** Definition of a path-can-finish activity.

The optional OrderNumber attribute on sdm:ActivityRef has no meaning here, and, when present, should be ignored by a receiving system.

## 5.2 Entry and Exit Criteria

Entry and exit criteria can be specified for each type of study-design structural element: activities, segments, cells, epochs, and study events. Zero or more entry criteria and zero or more exit criteria can be associated to any structural element. The entry and/or exit criteria may optionally reference the InclusionExclusionCriteria (see section 3.1.2). Here is an example of the XML for entry and exit condition specifications for an epoch and for an activity:

```
<sdm:EntryExitCriteria OID="CRIT_TRT_EPCH" Name="Treatment Entry Criteria"
  StructuralElementType="Epoch"
  StructuralElementOID="TREPOCH">
  <sdm:EntryCriteria>
    <sdm:IncludeInclusionExclusionCriteria/>
    <sdm:Criterion OID="CRIT_TRT_EPCH_01" Name="Treatment Entry Criterion 00"
      ConditionOID="COND_00"/>
  </sdm:EntryCriteria>
  <sdm:ExitCriteria>
    <sdm:Criterion OID="CRIT_TRT_EPCH_02" Name="Treatment Entry Criterion 01"
      ConditionOID="COND_01"/>
  </sdm:ExitCriteria>
</sdm:EntryExitCriteria>

<sdm:EntryExitCriteria OID="CRIT_ACG_ACT" Name="ECG Entry Criteria"
  StructuralElementType="Activity"
  StructuralElementOID="ACTDEF_ECG">
  <sdm:EntryCriteria>
    <sdm:Criterion OID="CRIT_ACG_ACT_01" Name="ECG Entry Criterion 00"
      ConditionOID="COND_02"/>
  </sdm:EntryCriteria>
  <sdm:ExitCriteria>
    <sdm:Criterion OID="CRIT_ACG_ACT_02" Name="ECG Entry Criterion 01"
      ConditionOID="COND_03"/>
  </sdm:ExitCriteria>
</sdm:EntryExitCriteria>
```

**Example 5.2:** Entry and exit criteria for an epoch, and for an activity.

The **EntryExitCriteria** element acts as a container for the entry and exit criteria to be associated with a given structural element. The OID and Name attributes (both mandatory) contain, respectively, a unique identifier and a short name for the container. The StructuralElementOID attribute on the EntryExitCriteria element references the structural element to which the contained criteria will be associated. The Type attribute on the EntryExitCriteria element (mandatory) can be one of: Activity, Segment, Cell, Epoch, StudyEvent.

An EntryExitCriteria element can contain zero or one Description elements, zero or one **EntryCriteria** element, and zero or one **ExitCriteria** element. Each occurrence of EntryCriteria or ExitCriteria acts as a grouping mechanism for all the actual criteria to be satisfied for entry or exit from the structural element to be permitted.

Each **Criterion** element within an EntryCriteria or ExitCriteria element must reference a condition via its ConditionOID attribute. The actual condition that must be satisfied is specified within an ODM ConditionDef declared in the appropriate location in the containing ODM document. The Criterion element's OID and Name attributes are mandatory and provide, respectively, a unique identifier and a short name for the criterion. Each EntryCriteria or ExitCriteria element may contain zero or more Criterion sub-elements.

All entry criteria must be met for the workflow to transition into the given structural element. Similarly, all exit criteria must be met for the workflow to transition away from the given structural element. Later subsections will describe in more detail the complete semantics to be honored by the workflow processor.

The optional **IncludeInclusionExclusionCriteria** element is an empty element without attributes, child elements or text content. Its presence within the EntryCriteria or ExitCriteria indicates that all criteria listed as InclusionExclusionCriteria for the study are criteria for the entry or exit of the referenced structural element. In such a case, all Inclusion/Exclusion criteria must also be met for the workflow transition to take place. The Inclusion/Exclusion criteria are met when all conditions referenced as inclusion criteria evaluate to true, and all conditions referenced as exclusion criteria evaluate to false. The absence of the IncludeInclusionExclusionCriteria element indicates that the criteria from the InclusionExclusionCriteria are not part of the entry criteria or the exit criteria of the given structural element.

## 5.3 Workflow Paths and Branching

The main purpose of a workflow definition is to provide for transitions between activities. Often, these transitions are not fixed, but are dependent on certain conditions. That is, a certain transition (branch) may be followed to the next activity only if a certain condition holds. If instead a different condition holds, a different branch will be followed. Such conditional branching enables, among other things, the modeling of repeating activities and groups of repeating activities, by defining a loop in the workflow, with a conditional transition (branch) at the loop's exit.

This section describes the branching mechanisms supplied to study designers by this specification.

### 5.3.1 Transition

A **Transition** element specifies the set of potential branches that could be followed after completion of a given activity. Transitions are associated with activities only. Transitions are not defined between segments, cells or epochs; a transition across such structures occurs only when a transition is made between activities contained in different segments, cells or epochs.

Transitions in workflows always override the natural or explicit order (given by OrderNumber) in structural element definitions (SegmentDef and StudyEventDef).

A Transition references the activity from which it originates via its mandatory attribute, SourceActivityOID. An activity can be referenced in this fashion exactly once; the runtime's behavior will be undefined in the case in which the same activity is referenced more than once via a Transition's SourceActivityOID attribute. So it is an error to have two or more Transition elements with the same value of SourceActivityOID within their parent Workflow element.

If an activity is not referenced by any Transition element's SourceActivityOID (i.e. it has no outgoing flow to a following activity), that activity is a 'dead end', except when it is in a workflow chain started from a trigger. A runtime may assume that this situation implies a default transition to the study finish, but this is not a required semantic – the runtime may take some other action such as raise an event/alert whose response may be manual. Ideally, validation of a study design would ensure that no dead ends exist before the design is inserted into a runtime workflow system such as a hospital planning system or a patient study calendar. The PathCanFinish element could also be applied to such an element as a signal to the design validator and/or runtime that the lack of outgoing transitions from that activity is an intentional part of the design rather than an error of omission.

A Transition is processed when all exit criteria for the source activity have been met. Other constructs, discussed in subsequent subsections, exist to describe the potential target/targets of a transition and the conditions under which a particular transition is followed.

The Transition element has mandatory **OID** and **Name** attributes providing a unique identifier and a short name for the transition.

### 5.3.2 Switch

Each Transition element must contain exactly one **Switch** element. A Switch defines a set of **TransitionDestination** elements (zero or more occurrences). The order of TransitionDestinations within each Switch is derived from the XML Document order, or an optional OrderNumber, attribute as described in section 2.4. Each TransitionDestination points to an activity through the TargetActivityOID attribute (mandatory) and also references an ODM ConditionDef through the **ConditionOID** attribute (mandatory).

When a TransitionDestination element is encountered, and its referenced condition evaluates to true, then the destination is to be followed.

```
<sdm:Transition OID="START_TRANS" Name="Transition from Study Start"
                SourceActivityOID="ACT_START">
  <sdm:Switch>
    <sdm:TransitionDestination OID="TRANS_START_ACT_VS01"
                              Name="To Vital Signs"
                              TargetActivityOID="ACT_VS01"
                              ConditionOID="COND_00" OrderNumber="1"/>
    <sdm:TransitionDestination OID="TRANS_START_ECG"
                              Name="To ECG"
                              TargetActivityOID="ACT_ECG"
                              ConditionOID="COND_01" OrderNumber="2"/>
  </sdm:Switch>
</sdm:Transition>
```

**Example 5.3.2.a:** A transition with multiple possible targets.

In addition, a Switch element may contain a **TransitionDefault** element (zero or one occurrence) as the last of its contained elements. The TransitionDefault specifies a target activity to which a transition must take place when that TransitionDefault is reached during Switch evaluation, i.e. when

none of the conditions for any supplied TransitionDestination elements evaluated to 'true'. A TransitionDefault may be processed as if it were a TransitionDestination whose condition always evaluates to 'true'. The TransitionDefault element has mandatory **OID** and **Name** attributes providing a unique identifier and a short name for the default transition. It has an optional ODM **Description** child element.

For example:

```
<sdm:Transition OID="START_TRANS" Name="Transition from Study Start"
  SourceActivityOID="ACT_START">
  <sdm:Switch>
    <sdm:TransitionDestination OID="TRANS_START_ACT_VS01"
      Name="To Vital Signs"
      TargetActivityOID="ACT_VS01"
      ConditionOID="COND_00" OrderNumber="1"/>
    <sdm:TransitionDestination OID="TRANS_START_ECG"
      Name="To ECG"
      TargetActivityOID="ACT_ECG"
      ConditionOID="COND_01" OrderNumber="2"/>
    <sdm:TransitionDefault OID="TRANS_START_DEFAULT"
      Name="Study Start Transition Default"
      TargetActivityOID="ACT_FINISH" />
  </sdm:Switch>
</sdm:Transition>
```

**Example 5.3.2.b:** A transition with a default target.

A Switch may contain only a TransitionDefault, which would mean that there is an “unconditional transition” from the source activity to the destination (a transition that will always be followed upon leaving the source activity). For example:

```
<sdm:Transition OID="START_TRANS" Name="Transition from Study Start"
  SourceActivityOID="ACT_START">
  <sdm:Switch>
    <sdm:TransitionDefault OID="TRANS_START_DEFAULT"
      Name="To From-Study-Start Default"
      TargetActivityOID="ACT_VS01" />
  </sdm:Switch>
</sdm:Transition>
```

**Example 5.3.2.c:** An unconditional transition.

It is possible, if no transition default is specified for a switch, for a switch’s evaluation to result in no transition to a destination, if no transition-destination conditions are met. The processor’s behavior in such a case is undefined; it is good design practice to ensure that either a default is in place for each switch, or that at least one transition destination will be followed given any set of inputs to the conditions to be evaluated.

## 5.4 Specifying Contingencies for Unplanned Events

A study designer may wish to specify that a particular activity is to be undertaken immediately if a given condition arises while at any point in the traversal through a given structural element. Triggers allow such contingencies to be specified.

### 5.4.1 Trigger

A **Trigger** denotes a divergence to a new path which can be started at any point during a study participant's path within the referenced structural element.

For example:

```
<sdm:Trigger OID="FEVER_TRIGGER" ConditionOID="FEVER_DISCOVERED"
  Name="Fever Discovery"
  StructuralelementOID="TREPOCH"
  StructuralelementType="Epoch">
  <sdm:Switch>
    <sdm:TransitionDefault OID="FEVER_TRIGGER_TRANS"
      Name="Transition on Fever Discovery"
      TargetActivityOID="ACTDEF_ASSESSFEVER_01" />
  </sdm:Switch>
</sdm:Trigger>
```

**Example 5.4.1.a:** An unconditional trigger.

In the example above, when fever is discovered (as described by the Condition with OID "FEVER\_DISCOVERED") during the treatment epoch (TREPOCH), the activity with OID "ACTDEF\_ASSESSFEVER\_01" must be executed. If, however, fever is discovered outside the treatment epoch, the trigger is not applicable.

Also here, there may be different paths once the trigger is fired, each of which depends on a particular condition. For example, when fever is discovered (Trigger with condition OID "FEVER\_DISCOVERED"), there may be a separate path for the case of high fever (>100°F, 38.5°C), and the default path for all other cases:

```
<sdm:Trigger OID="FEVER_TRIGGER" ConditionOID="FEVER_DISCOVERED"
  Name="Fever Discovery"
  StructuralelementOID="TREPOCH"
  StructuralelementType="Epoch">
  <sdm:Switch>
    <sdm:TransitionDestination ConditionOID="HIGH_FEVER_DISCOVERED"
      Name="Transition when subject has very high fever"
      OID="TRANS.HIGH_FEVER_TRIGGER"
      TargetActivityOID="ACTDEF_ASSESS_HIGH_FEVER"/>
    <sdm:TransitionDefault OID="FEVER_TRIGGER_TRANS"
      Name="Transition on other Fever Discovery"
      TargetActivityOID="ACTDEF_ASSESSFEVER_01" />
  </sdm:Switch>
</sdm:Trigger>
```

**Example 5.4.1.b:** A conditional trigger.

The path initiated by the trigger can have further activities that follow the ones defined by the Trigger element. For example, the activity "ACTDEF\_ASSESS\_HIGH\_FEVER" may be followed by other conditional or unconditional activities. These paths are described by normal "Transition" elements.

A path traversal initiated by a Trigger may return to the 'normal' workflow via any transition specification to an activity in that workflow. In cases where the point of return cannot be anticipated at design time, an activity at the end of a triggered path may be marked as PathCanFinish, which can be used to signal the runtime that some intervention or calculation may be needed to determine the point of return from the trigger path. For example, in case of a minor adverse event with no consequences for the further course of the study, the last activity in the chain triggered by that event may be marked as "PathCanFinish" (i.e. referenced under the "PathCanFinish" element), whereas a serious adverse event with consequences for the course of the study will usually have some path to the normal workflow, such as a transition to the first activity in the "washout" part, or even to the activity that was marked as "study finish".

## 5.5 Processing Transitions between Activities

The combination of entry/exit criteria and branching allows for very powerful combinations to be created. The general transition semantics that the runtime execution would need to provide are:

- On the source activity that has completed:
  - Check that exit criteria have been met.
  - Check whether the exit criteria are met for all structural elements that contain the activity.
  - Determine the transition that has the completed activity as its source.
  - Process that transition's switch to determine the transition to be followed, and the target activity of that transition.
- For the transition path to be followed:
  - Check whether the entry criteria are met for all structural elements that contain the target activity.
  - Check the entry criteria for the target activity are met.
  - If all the above are fulfilled then the transition is complete, and the study participant's path can proceed to the target activity.

## 5.6 Workflow Transitions between Visits, Cells, Epochs and Segments

SDM-XML provides explicit workflow transitions only between Activities. Transitions between StudyEvents (i.e. visits) that otherwise do not contain Activities appropriate for use as the source or target of a workflow transition can be modeled by defining a "visit start activity" and "visit finish activity", and adding these activities to the StudyEvent definition. Workflow transitions then could be defined that reference these activities. The same technique may be used as needed to model workflow transitions between other types of structural containers (segments, epochs, and cells).

# 6 Timing

Timing constraints, like structural definitions and workflow definitions, are declared in their own sub-section of an SDM-XML document, within an element named **Timing**. Timing constraints may apply either to activities or to workflow transitions. Note, however, that this relationship is one-way – elements declared in the structural or workflow areas of the document never reference timing elements.

## 6.1 Timing Concepts

This section introduces core concepts employed in the SDM-XML constructs for describing timing constraints. Later sections give specific examples of the usage of these concepts.

### 6.1.1 Timing Windows

In specifying a timing constraint, a study designer can specify an ideal time for the elapsed time between activities or workflow transitions. In addition, designers may specify a “pre-window” and a “post-window” around this ideal time – durations before and after the ideal time, within which the timing constraint can be considered to be satisfied. Different values may be provided for the pre- and post- durations.

TimepointPreWindow and TimepointPostWindow are always specified as positive durations. The fact that the TimepointPreWindow start will occur before the target timepoint is implicit to the attribute.

### 6.1.2 Timepoint Granularity

The TimepointGranularity attribute allows the designer to override the timepoint target and pre/post windows to widen the scope of a timing window. The following are the available values:

- PY – it is allowed for the activity’s timepoint to happen anytime in that year.
- PM – it is allowed for the activity’s timepoint to happen anytime in that month.
- PD – it is allowed for the activity’s timepoint to happen anytime in that day.
- PTH – it is allowed for the activity’s timepoint to happen anytime in that hour.
- PTM – it is allowed for the activity’s timepoint to happen anytime in that minute.
- PTS – it is allowed for the activity’s timepoint to happen anytime in that second.

In the preceding list, the activity’s timepoint is identified by the Type attribute. As an example, consider an activity A that has a default transition to activity B. The TransitionDestination has a timing associated with it that has the following values:

- Type: FinishToStart
- TimepointRelativeTarget: PT48H
- TimepointGranularity: PD

This means that the start of activity B should happen at any point in the day that is 48 hours after activity A finishes. For more information about the timing Type attribute, see the section “Timing Types”.

### 6.1.3 Timing Types

When considering activities relative to one another, it is natural to think of one activity occurring a period of time *after* the other one finishes. However, other types of relationships can exist and the Type attribute embodies this. Valid values for the attribute are:

- StartToStart – the start of the subsequent activity should be relative to the start of the previous activity.
- StartToFinish – the finish of the subsequent activity should be relative to the start of the previous activity.
- FinishToStart – the start of the subsequent activity should be relative to the finish of the previous activity.
- FinishToFinish – the finish of the subsequent activity should be relative to the finish of the previous activity.

FinishToStart is the default timing type, and is applied when the Type attribute is omitted.

## 6.1.4 SubsequentSchedulingBasis

By default, activities that are related to another activity, either by a TransitionTimingConstraint or RelativeTimingConstraint, are scheduled relative to the planned time for the given activities. However, a designer may also wish that timing on subsequent activities be applied relative to the actual time that an activity actually takes place during study execution.

The optional attribute SubsequentSchedulingBasis on the elements mentioned above provide a way to do this. The valid values are:

- Planned – during the course of the study execution, the subsequent activity should be scheduled relative to the planned time of the predecessor activity. This is the default, and is applied if the attribute is not specified.
- Actual – during the course of study execution, the subsequent activity should be scheduled relative the actual time of the predecessor activity (once the actual time is known).

Note that this attribute is not available on the AbsoluteTimingConstraint.

## 6.2 Absolute Timing Constraints

**AbsoluteTimingConstraint** is used to limit when an activity can take place during a given time interval, or to specify an exact date and time as the ideal timing for an activity.

The example below shows an AbsoluteTimingConstraint element that specifies an ideal time as 10:30am Eastern Standard Time, but which permits the referenced Activity to take place one hour before or after that time.

```
<sdm:AbsoluteTimingConstraint OID="TC_A_02"  
    Name="Ambulatory ECG Timing"  
    ActivityOID="ACTDEF_AMBECGPLACED_01"  
    TimepointTarget="-----T10:30:00-05:00"  
    TimepointPreWindow="PT1H"  
    TimepointPostWindow="PT1H"/>
```

**Example 6.2:** An absolute timing constraint.

To specify an exact date and time as the ideal timing for an activity, give a full date and time as the value for AbsoluteTimingConstraint's TimepointTarget attribute.

## 6.3 Relative Timing Constraints

**RelativeTimingConstraint** is very similar to TransitionTimingConstraint except that it references the predecessor and successor activities directly. It allows the designer to specify the timing of any two activities relative to one another by the start or finish time of either activity.

The example below illustrates a relative timing constraint between 2 activities that are not connected by a transition and could occur at different points in the study. It shows that the ADAS-Cog test should be performed 56 days after the finish of the study participant randomization. However it also is allowed to happen 2 days before, or 5 days after the target time calculated by the execution runtime.

```

<sdm:RelativeTimingConstraint OID="TC_R_02"
    Name="Constraint between randomization and cognition test"
    PredecessorActivityOID="ACTDEF_PATIENTRANDOMIZED"
    SuccessorActivityOID="ACTDEF_ADASCOGTEST_01"
    Type="FinishToStart"
    TimepointRelativeTarget="P56D"
    TimepointPreWindow="P2D"
    TimepointPostWindow="P5D"/>

```

**Example 6.3.a:** A relative timing constraint.

Note that in the example above that the actual clock-time of the window may come into play. If the patient randomization activity finished at 2:00pm then the allowable window for the ADAS-Cog test may be calculated to start at 2:00pm 54 days later and close at 2:00pm 61 days later.

The example below is a modification of the previous one and illustrates the use of **TimepointGranularity**. It would modify the window for the ADAS-Cog test to start anytime on the 54<sup>th</sup> day after the patient randomization and close anytime on the 61<sup>st</sup> day.

```

<sdm:RelativeTimingConstraint OID="TC_R_02"
    Name="Constraint between randomization and cognition test"
    PredecessorActivityOID="ACTDEF_PATIENTRANDOMIZED"
    SuccessorActivityOID="ACTDEF_ADASCOGTEST_01"
    Type="FinishToStart"
    TimepointRelativeTarget="P56D"
    TimepointGranularity="PD"
    TimepointPreWindow="P2D"
    TimepointPostWindow="P5D"/>

```

**Example 6.3.b:** A relative timing constraint, with per-day granularity.

The Type attribute specifies the type of dependency between the source and destination activities. The default (and likely most common) should be considered finish-to-start ("FinishToStart"), which means that the timing constraint applies to the interval between the finish of the source activity and the start of the destination activity.

## 6.4 Transition Timing Constraints

**TransitionTimingConstraint** is associated with a TransitionDestination element. It provides the study designer with the ability to add relative timing information to a transition using the activity referenced by the ActivityTransition as the source timing anchor.

The TimepointRelativeTarget attribute is a duration that defines the ideal interval between the source activity [anchor] and the destination activity. The TimepointPreWindow and TimepointPostWindow attributes are also durations that identify the window around the ideal interval where it is allowed for the activity to take place. The time intervals are based on the ISO-8601 date time format. (See [http://en.wikipedia.org/wiki/ISO\\_8601#Time\\_intervals](http://en.wikipedia.org/wiki/ISO_8601#Time_intervals) for a general explanation of ISO8601 time-interval notations. The full specification can be purchased at [http://www.iso.org/iso/catalogue\\_detail?csnumber=40874](http://www.iso.org/iso/catalogue_detail?csnumber=40874)).

The example below shows a transition timing constraint between two transitions. For workflow, the transition element points to the ambulatory ECG placement as the source and the ambulatory ECG removal as the [default] destination for the transition. The TransitionTimingConstraint element adds

a timing dimension to this by referencing the transition destination – in this case the TransitionDefault element. The timing information specifies that the ECG removal activity should happen between 23-25 hours after the placement, with the ideal being 24 hours.

```
<sdm:TransitionTimingConstraint OID="TC_T_02"  
    Name="Example constraint on a transition"  
    TimepointRelativeTarget="PT24H"  
    TimepointPreWindow="PT1H"  
    TimepointPostWindow="PT1H"  
    TransitionDestinationOID="TRANSDEST_01"/>
```

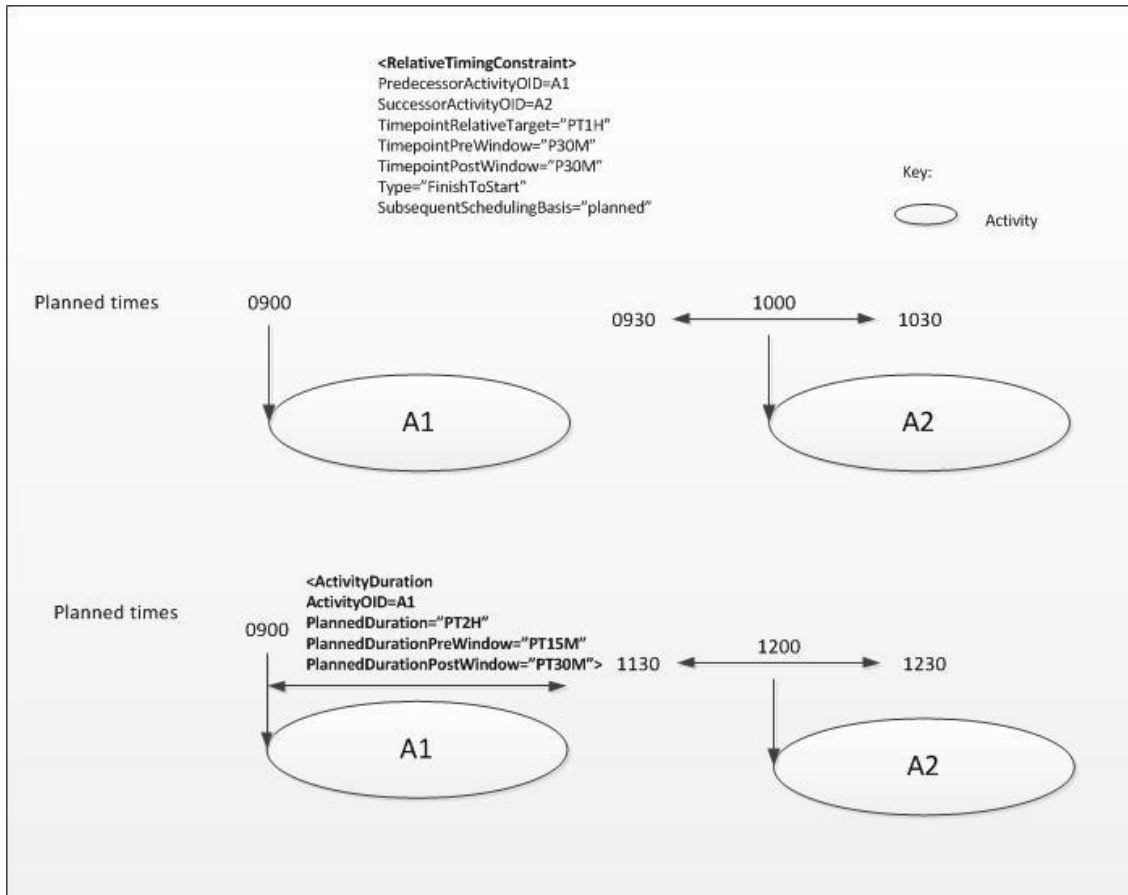
**Example 6.4:** A timing constraint on a transition.

## 6.5 Activity Durations

In a study design, it may be useful to designate how long a given activity is expected to take. A person may say “Activity X normally takes two hours. It may be completed in an hour and forty-five minutes on a good day or two and a half hours on a bad day”. The ActivityDuration element allows a designer to express this in the SDM-XML. In addition to this, it allows a design application to identify potential issues with timing constraints and an execution engine to more accurately project when activities should take place.

The ActivityDuration element refers to the activity in question using the ActivityOID attribute. It has a PlannedDuration to provide the normal length of time and PlannedDurationPreWindow, PlannedDurationPostWindow to provide for an acceptable window around that ‘normal’ time.

Consider the scenarios shown in this diagram:



In the first example of the above diagram, no ActivityDuration is specified. Assuming the activity A1 was scheduled to start at 0900, A2 would be expected to start at 1000. In the second example, A1 has been defined to have a duration of 2 hours and is scheduled to start at 0900. The planned finish is then 1100 so the RelativeTimingConstraint can be applied to that (due to the FinishToStart type). A2 would then be scheduled to start at 1200 based on the planned duration of A1.

## 6.6 Evaluation of Timing Constraints

Timing constraints define target time for activities and, optionally, a window around which the activity may take place. There may be zero or more timing constraints on an activity, each of which may define a different target time and windows.

A study execution runtime must be able to reconcile multiple applicable timing constraints to identify:

- If the set of all windows does not yield an "intersection" duration within which all constraints may be satisfied. If a set of constraints does not yield at least one valid moment in time during which an activity or transition may take place, then the runtime must raise an error, either at design time or during execution, depending on when and how the problem was introduced.
- In the case of multiple constraints applicable to the same activity or transition, the runtime can apply its own heuristic to determine the aggregate "ideal" time. Examples of such heuristics include taking the mean of the aggregate window's upper and lower bounds, and taking the median ideal time of all contributing timing constraints and using it if the median ideal occurs within the aggregate window.

## 6.7 Timing Constraints on Visits, Cells, Epochs and Segments

SDM-XML provides explicit timing constraints only on Activities and on workflow transitions between activities. Timings between StudyEvents (i.e. visits) can be modeled by defining a "visit start activity" and "visit finish activity", neither of which would require data collection, and adding these activities to the study event. A timing constraint then could be defined that applies to these activities, as needed. The same technique may be used to model timing constraints between other types of structural containers (segments, epochs, and cells).

# 7 Element Reference

## 7.1 ODM V1.3.1 Elements Used by Study Design

As this is an extension of the CDISC ODM V1.3.1 schema, elements of that schema are referenced in the study design schema. For a full explanation of ODM elements, refer to the ODM V1.3.1 specification.

## 7.2 Study Design-Specific Elements

This specification package includes an HTML reference to elements and attributes declared in the SDM-XML namespace. From the default unzip of the specification package, this HTML reference can be found [here](#).

# 8 Examples

This specification package includes several example SDM-XML files. The examples folder of the specification package contains a valid SDM-XML file for each XML source example given in this document. In addition, the examples/Misc folder contains a larger example file (LZZTtrial.xml) that partially defines a study, making use of a wide range of constructs described in this specification. These examples exist only to demonstrate the syntax and semantics of SDM-XML, and are not intended to fully describe a real-world study.

The examples/Misc/Display folder contains a version of the LZZTtrial.xml file that leverages an XSL (Extensible Stylesheet Language) file to enable translation of the SDM-XML data to HTML, viewable in web browsers. The display stylesheet serves only as an example of the possibilities for display of SDM-XML information. Consumers of SDM-XML are free to use the provided stylesheet, or any other display mechanism of their choosing, for visualization of SDM-XML study designs.

# 9 XML Schema for SDM-XML

The XML Schema for Study Design extends ODM using the same pattern as that employed by the CRT-DDS standard. There is a root schema, an extension schema (which defines the extensions to existing ODM elements such as Protocol), and an 'ns' schema, which defines those elements and attributes specific to study design.

Further, the Study Design Model's 'ns' schema includes separate schema documents for structural information, workflow information, and timing information. The default unzip of the specification package includes all Study Design, ODM and common schema files required to validate SDM-XML documents.

## 10 Acknowledgments

This specification was developed by the CDISC SDM-XML Technical Committee:

- Jan Kratky, SAS (Editor)
- Andrew Fowler, DevFrame (Section Editor: Structure)
- Jozef Aerts, XML4Pharma (Section Editor: Workflow)
- Peter Villiers, SAS (Section Editor: Timing)
- Peter Abramowitsch, Medidata
- Joyce Hernandez, Merck
- Gary Walker, Quintiles
- Mary Lenzen, Octagon Research Solutions
- Diane Wold, GlaxoSmithKline

Thanks also to Sam Hume (AstraZeneca), Kevin Burges (Formedix) and Sally Cassells of the CDISC XML Technologies Governance Committee, as well as to all who took time to review and comment on this specification.