

SWIM-INFO-007 Information definition concepts

Page Table of Content

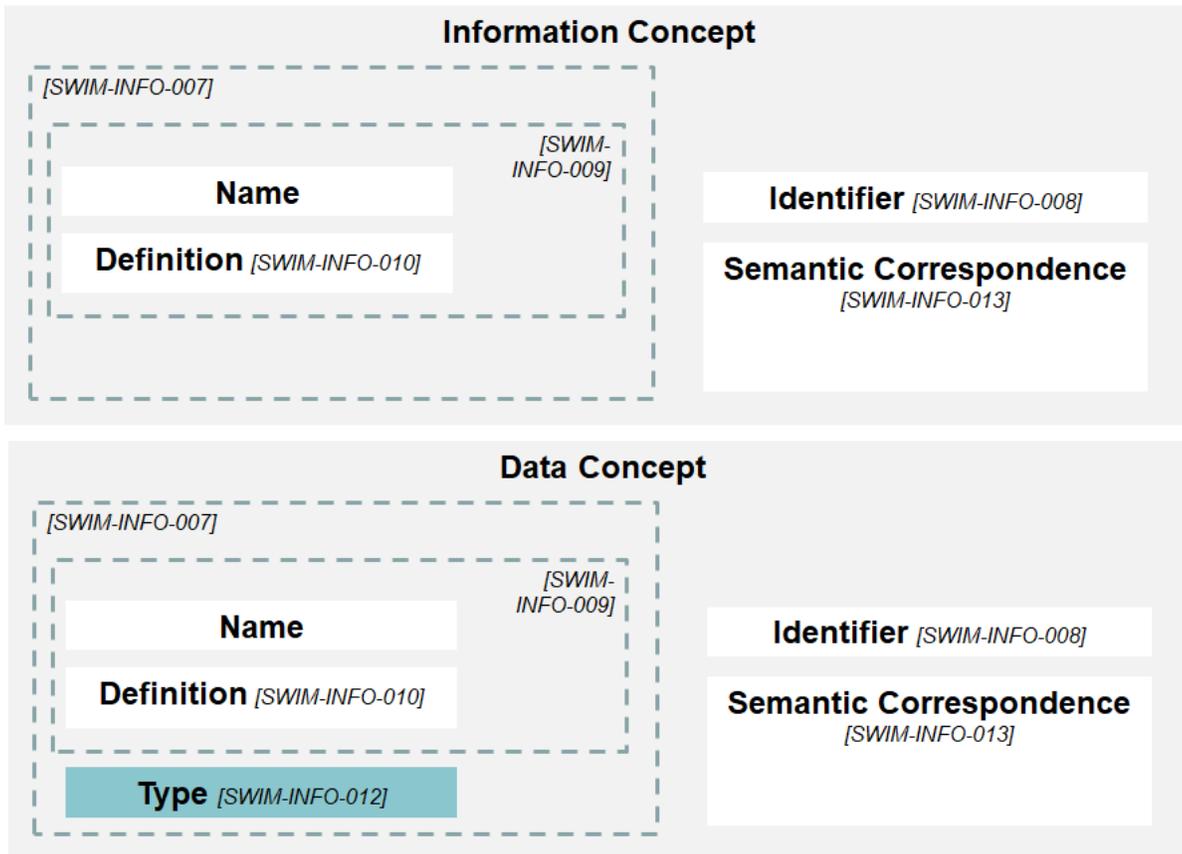
- Requirement
- Guidance
 - Verification Support
 - Completeness
 - Consistency
 - Correctness
 - Examples
 - Concept name
 - Concept definition
 - Concept data type

Requirement

Title	Information definition concepts
Identifier	SWIM-INFO-007
Requirement	An information definition shall specify concepts by providing, at a minimum: <ul style="list-style-type: none">• the name of the concept;• the definition of the concept; and• for data concepts only, the data type of the concept.
Rationale	This requirement gives the minimum details needed about a concept (information concepts and data concepts) in order to apply the other requirements such as the ability to establish a semantic correspondence between concepts based on the definitions of the concepts.
Verification	Completeness, Consistency, Correctness
Examples/Notes	<p><i>Note:</i> This requirement concerns information concepts and data concepts.</p> <p><i>Note:</i> This requirement is deliberately generic to allow multiple notations and modelling techniques. For example, concepts may be represented as classes in UML, elements in XML Schemas or as rows in a spreadsheet.</p> <p><i>Note:</i> As the requirement sets out the minimum needed, there is nothing to stop an information definition from including more details such as the relationships between concepts and constraints on the concepts. Furthermore, concepts may be contained in other concepts.</p>
Level of Implementation	<i>Mandatory</i>

Guidance

This requirement applies to [information concepts](#) and [data concepts](#). However, the requirement treats them differently. This difference is visualised below. In effect, a data concept needs to have an associated [data type](#) (coloured below). *SWIM-INFO-012* places additional requirements on the usage of data types. The method for establishing the semantic correspondence for the different concepts is also different (see *SWIM-INFO-016* and *SWIM-INFO-017*).



The figure below is used to show how the different concepts would appear in UML. In this case, the example is taken from AIXM 5.1.1. The *AirportHeliport* feature is an information concept as it has a name and a definition. It contains attributes such as *locationIndicator/CAO* which are data concepts - they have names, definitions and data types (*Code/CAOType*).

«feature»
AirportHeliport

- + abandoned: CodeYesNoType
- + altimeterCheckLocation: CodeYesNoType
- + certificationDate: DateType
- + certificationExpirationDate: DateType
- + certifiedICAO: CodeYesNoType
- + controlType: CodeMilitaryOperationsType
- + dateMagneticVariation: DateYearType
- + designator: CodeAirportHeliportDesignatorType
- + designatorIATA: CodeIATAType
- + fieldElevation: ValDistanceVerticalType
- + fieldElevationAccuracy: ValDistanceVerticalType
- + landingDirectionIndicator: CodeYesNoType
- + locationIndicatorICAO: CodeICAOType
- + lowestTemperature: ValTemperatureType
- + magneticVariation: ValMagneticVariationType
- + magneticVariationAccuracy: ValAngleType
- + magneticVariationChange: ValMagneticVariationChangeType
- + name: TextNameType
- + privateUse: CodeYesNoType
- + referenceTemperature: ValTemperatureType
- + secondaryPowerSupply: CodeYesNoType
- + transitionAltitude: ValDistanceVerticalType
- + transitionLevel: ValFLType
- + type: CodeAirportHeliportType
- + verticalDatum: CodeVerticalDatumType
- + windDirectionIndicator: CodeYesNoType



The specification requires only a minimum amount of details for the concepts. This reflects the focus on semantic interoperability. However, more details such as multiplicity and data value ranges can be added to the information definition. Indeed, the Specification for Service Descriptions, *SWIM-SERV-022* requires more.



The note to the requirement in the Specification mentions that it is possible to include "*more details such as the relationships between concepts*". This is often done by adding associations in UML models, or by using attributes to link different concepts. The Specification does not recommend one technique over the other. Indeed, the Specification does not mention techniques at all as the manner of adding relationships differs depending on the format used for the information definition.

Verification Support

Completeness

Check that:

- The information definition contains a name for each of its information concepts and data concepts.
- The information definition contains a definition for each of its information concepts and data concepts.
- The information definition contains a data type for each of its data concepts.

Consistency

Check that:

- There are no duplicated concepts in the same namespace within the information definition.
- The concepts within the information definition are consistent with each other i.e. that concepts do not contradict each other.

[] The concepts within the information definition are consistent with their reference material.

Notes:

1. Reference material includes e.g. any ISO standards used in the information definition.

Correctness

Check that:

[] Each concept's name, definition and data type are correct i.e. that they actually represent what they should.

Examples

Concept name

The example below shows how a concept name could appear in an XML schema. It simply uses the name attribute of an element.

Example of SWIM-INFO-007

```
<xs:element name="tobt">
  ...
</xs:element>
```

Concept definition

The example below also shows how a concept's definition appears in an XML schema. It makes use of the annotation element within the XML schema definition language.

Example of SWIM-INFO-007

```
<xs:element name="tobt">
  <xs:annotation>
    <xs:documentation>
      <definition>The Target Off-Block Time value to be set. TOBT is the time that an operator / handling agent estimates that an aircraft will be ready, all doors closed, boarding bridge removed, push back vehicle present, ready to start up / push back immediately upon reception of clearance from the TWR.</definition>
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

Concept data type

The example below shows two ways to capture the data type information in an XML schema. The first states that the "ICAOAerodromeLocationIndicator" element is of type "xs:string". The second shows how an XML simpleType (using the same example) can be restricted to be a "xs:string". These examples show different techniques for creating schemas but are not concerned with which is the better technique.

Example of SWIM-INFO-007

```
<xs:element name="ICAOAerodromeLocationIndicator" type="xs:string">
  ...
</xs:element>

<xs:simpleType name="ICAOAerodromeLocationIndicator">
  ...
  <xs:restriction base="xs:string">
    ...
  </xs:restriction>
  ...
</xs:simpleType>
```