Understanding and recording mappings

Purpose

This guidance is provided in order to allow you to **better understand** mappings from an information definition to the AIRM and **how to record them** in the information definition. The guidance applies to:

- SWIM-INFO-016 Mapping of information concepts
- SWIM-INFO-017 Mapping of data concepts
- SWIM-INFO-018 Additional traces to clarify the mapping

The different forms of semantic correspondence are outlined in SWIM-INFO-014 Forms of semantic correspondence. This guidance applies to mappings (the first option in the requirement). Mappings contain one or more traces. Therefore, this guidance covers the different types of traces.

Different types of traces

The table below outlines the names, definition and relevant requirement to be used for the different types of trace that constitute a mapping statement. The names are inspired by the words from the SWIM Information Specification's requirements. This approach makes it clear which requirement is being satisfied by the trace.

Requirement	Trace name	Definition
SWIM-INFO-016 Mapping of information concepts	"information concept" trace	trace from the information concept in the information definition to the AIRM concept that has an equivalent or wider meaning
SWIM-INFO-017 Mapping of data concepts	1. "data concept" trace 2. "data type" trace	1. trace from the data concept in the information definition to the AIRM concept that has an equivalent or wider meaning 2. trace to the data type in the AIRM that has an equivalent or wider meaning
SWIM-INFO-018 Additional traces to clarify the mapping	"narrowing" trace	trace to an AIRM concept to fully describe the narrowing of the concept being mapped

Source and target of traces

The Interoperability Architecture provides good guidance on the best place to start when looking to establish a mapping. Basically, the best place to start is usually the adjacent box within the grid.

The best start point when identifying a suitable AIRM concept depends on the type of information definition being traced. This can also give an indication on the type of trace to be used. The table below gives some general guidance on this.

Type of information definition	General guidance

<table>
<tr><td>information exchange requirements</td><td>The best place to start in order to identify a suitable AIRM concept is the AIRM Conceptual Model. However, information exchange requirements can vary in the level of detail included. Therefore, if no suitable AIRM concept is found in the AIRM Conceptual Model, the AIRM Logical Model may be useful.

For the most part, it is expected that information exchange requirements involve "information concept" traces and so fall under requirement 16.

Narrowing traces (requirement 18) can be added as needed. It is usual that these trace to the same part of the AIRM as the "main" trace.

ⓘ The specification doesn't rule out tracing to the AIRM Contextual Model but this is not a good practice.</td></tr>
<tr><td>service payload /

information exchange model</td><td>Service payloads and information exchange models can include concepts that are of different level of granularity. For example, they may contain standardised "messages" such as NOTAM and METAR. They also contain concepts such as "Aerodrome" or "Airspace". These in turn may have attributes/properties such as the "ICAO location indicator".

Although it is difficult to give generic advice that is applicable in all cases, the following guidance is applicable:

• Standardised "messages" are captured in the AIRM Conceptual Model. Mapping of such messages tend to be involve "information concept" traces and so fall under requirement 16. If no suitable AIRM concept is found in the AIRM Conceptual Model, the AIRM Logical Model may be used.

⚠ It is an AIRM design decision to allow messages to be added at the service level. The AIRM Logical Model does not impose any message structure. However, the existence of the standardised messages is captured as part of the operational language in the AIRM Conceptual Model.

• The best place to start when mapping concepts that have no associated data type is the AIRM Logical Model. Concepts of this nature may, for example, be modelled as classes in UML models. These tend to involve "information concept" traces as they do not have a "data type" associated with them. This means that they fall under requirement 16 . If no suitable AIRM concept is found in the AIRM Logical Model, the Conceptual Model may be used.
• Attributes/properties will have a "data type" and therefore fall under requirement 17, requiring a "data concept" trace and a "data type" trace. If no suitable AIRM concept is found in the AIRM Logical Model, the Conceptual Model may be used.

⚠ The AIRM has internal traces to ensure consistency between the AIRM Conceptual Model and the AIRM Logical Model.

Narrowing traces (requirement 18) can be added as needed. It is usual that these trace to the same part of the AIRM as the "main" trace.

ⓘ The specification doesn't rule out tracing to the AIRM Contextual Model but this is not a good practice.</td></tr>
</table>

Reading order of traces

The standard requires multiple traces to be added to a mapping. The general reading order is:

1. "information concept" trace
2. "narrowing" traces (0..*)

or

1. "data concept" trace
2. "data type" trace (1)
3. "narrowing" traces (0..*)

All traces have an AND relationship.

ⓘ The following rules apply to the traces:

- The root trace is mandatory. This is either an "information concept" trace or a "data concept" trace.
- A "data type" trace is mandatory when the root trace is a "data concept" trace.
- "Narrowing" traces cannot exist in their own right.

The number of traces

The important thing when creating a mapping is to add sufficient traces to ensure that the semantics are understood. There is no need to add further traces.

If you find that too many traces are required to ensure that the semantics are understood, there may be a problem somewhere in fully understanding the meaning of a concept. In that case a change may be needed to the information definition and/or the AIRM.

Level of semantic correspondence

Advanced users may like to add extra detail concerning the degree of semantic correspondence achieved. The skos standard calls this the "semantic relation" between concepts.

The requirements talk about mapping to the concept with "equivalent or wider meaning". The table below outlines the skos sematic relation term that can be used in order to make the level of semantic correspondence explicit. It also contain the equivalent terms that were used in SESAR.

The skos names are preferred. Skos has rich support in semantic technologies.

However, existing SESAR documents use different names and it is important that readers can understand those traces - the table therefore includes those.

Definition being traced to is...	Skos annotations that can make this more explicit	Term used in SESAR documents
Equivalent	**skos:exactMatch:** is used to link two concepts, indicating a high degree of confidence that the concepts can be used interchangeably across a wide range of information retrieval applications. **skos:closeMatch:** is used to link two concepts that are sufficiently similar that they can be used interchangeably in some information retrieval applications.	**exactCopy**: Definition of concepts in the information definition and the AIRM are exact copy of each other. **syntacticallyEqual**: Definitions are only different due to syntax corrections (grammar, spelling) but are otherwise equivalent. **rewritten**: The definition of the concept in the information definition has been rewritten to reflect information definition specificity. However, the meaning is the same, i.e. the definition still describes exactly the same concept as the AIRM.
Wider	**skos:narrowMatch:** used to state a hierarchical mapping link between two concepts.	**specialised**: The definition in the information definition is a special case of the definition found in the AIRM.

We only need narrowing traces if the main trace is "specialised" or "narrowMatch"

Traces cannot be annotated as "generalised" as this breaks the requirement.

Annotating traces

It is possible to add further notes to the mapping (the container for one or more trace). This comes in handy when e.g. tracing legacy interfaces that have data type constraints leading to loss of Information.

Recording traces in XSD

The table below gives two alternatives for recording the traces in XSD.

Using element names is the preferred option as it can be used more easily in rules. The element name contains semantic hints even if the attributes are not added.

However, the attribute option is also supported as there are a lot of traces developed that do not use element names. Support for this option should be deprecated in the future.

Trace name	Element name	Attribute

"information concept" trace	<informationConceptTrace>	<trace keyword="informationConceptTrace>
1. "data concept" trace 2. "data type" trace	<dataConceptTrace> <dataTypeTrace>	<trace keyword="dataConceptTrace> <trace keyword="dataTypeTrace>
"narrowing" trace	<narrowingTrace>	<trace keyword="narrowingTrace>

XSD Example

If we apply the guidance above we get the following in XML Schema notation.

```
<xs:annotation>
  <xs:documentation>
    <semanticCorrespondence>
      <mapping>
        <note>loss of info because of legacy</note>
        <informationConceptTrace semanticRelation="specialised">-AIRM unique identifier-<
/informationConceptTrace>
        <narrowingTrace>-AIRM unique identifier-</narrowingTrace>
      </mapping>
    </semanticCorrespondence>
  </xs:documentation>
</xs:annotation>
```

or:

```
<xs:annotation>
  <xs:documentation>
    <semanticCorrespondence>
      <mapping>
        <note>loss of info because of legacy</note>
                <trace type="informationConceptTrace" semanticRelation="specialised">-AIRM unique identifier-
</trace>
        <trace type="narrowingTrace">-AIRM unique identifier-</trace>
      </mapping>
    </semanticCorrespondence>
  </xs:documentation>
</xs:annotation>
```