Australasian Transport Research Forum 2010 Proceedings
29 September – 1 October 2010, Canberra, Australia
Publication website: http://www.patrec.org/atrf.aspx

Integrating global aviation data with temporal and spatial weather information

Laura Irina Rusu [1], Wenny Rahayu[2], Torab Torabi[3,]

Florian Puersch[4], Andrew Taylor Harris[5], Karl Reed[6]

[1, 2, 3, 4, 6] Department of Computer Science and Computer Engineering,

La Trobe University, Bundoora VIC 3086

[5] AirServices Australia - Melbourne, Tullamarine, VIC 3043

Email for correspondence: L.Rusu@Latrobe.edu.au

Abstract

During the last few years, the fast and accurate exchange of information between players in various domains has become increasingly critical. This is especially true in transportation, where the rapid and correct decisions based on high-quality timely data would have not only financial implications (e.g. saving cost of fuel) but it could primarily relate to the safety of people. Messages are naturally exchanged using different standards, e.g. in aviation a weather message would differ from an airport notification message. Hence, it is often a challenge for the end user to be able to integrate and analyse the information received 'on the fly', especially when there is a temporal or spatial overlapping of information between messages, or when there are numerous messages.

XML (eXtensible Markup Language) is the format of choice for describing many standards and exchanging messages; therefore, our research proposes a generic approach to integrate XML messages which have temporal and spatial features. This approach gives the user the ability to apply 'time' and 'space' filtering on the resulting integrated data, in order to get a synchronised or an area-focused view of the events. The case study in this paper employs two XML standards used by the aviation industry, namely Aeronautical Information Exchange Model (AIXM) and Weather Information Exchange Model (WXXM). We show that the result of 'time' and 'space' filtering on the integrated data can help faster decision making in weather related aviation data analysis, for example determining the airspaces affected by a volcanic ash cloud.

1. Introduction

During the last few years, many XML standards have emerged, to control the way how data is exchanged between players, in various domains. Some of these standards include a *temporal aspect* (i.e. they allow for the exchange of dynamic data, which is only valid for certain periods of time) or a *spatial aspect* (i.e. they allow to specify a spatial/geographical delimitation of data applicability). Some XML standards include both these aspects.

When messages conforming to different XML standards are exchanged between the same players, it becomes highly critical to integrate them, because:

(i) it would dramatically decrease the time required by the users to interpret separate messages and manually integrate the information;

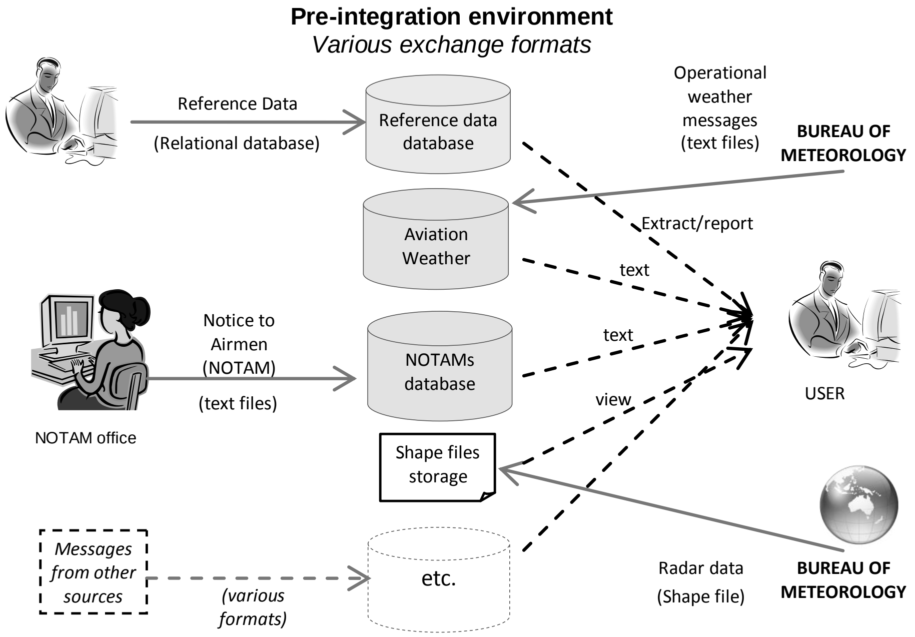
(ii) it would eliminate the risks inherently associated when the manual/ad-hoc integration is required, and

(iii) it might help in revealing knowledge about the combined data which is otherwise difficult to identify.

The integration of different standards is not a straightforward task, especially when the separate standards (schemas) have temporal and/or spatial features. Moreover, integration of temporal and spatial features is especially critical in those domains where timely and accurate decisions are extremely important (for example in transportation or health).

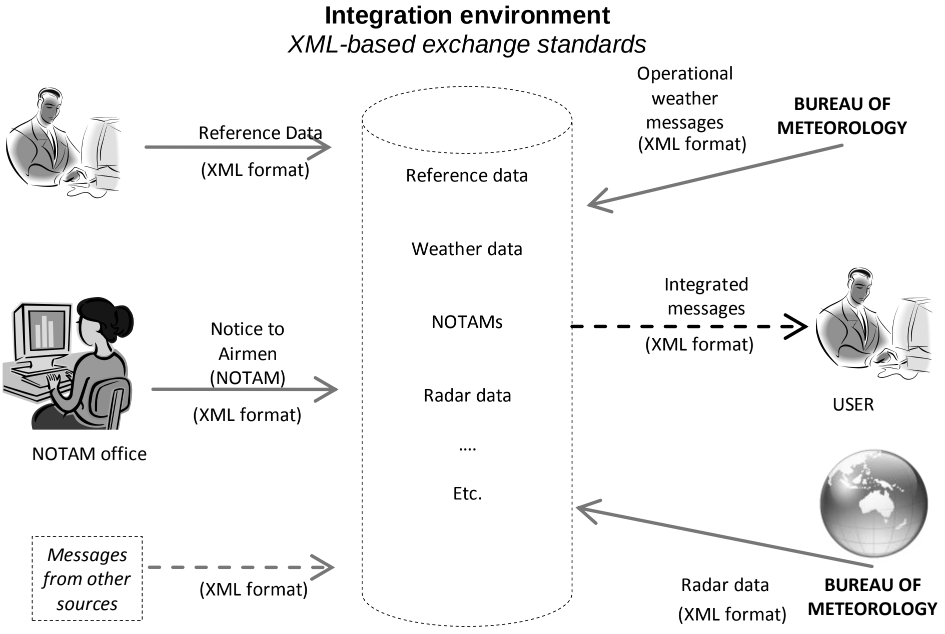
Figure 1 shows a visual representation of few types of aviation-related messages which are currently received, processed and utilised by AirServices Australia (ASA).

Figure 1: Visual representation of temporal and spatial messages in pre-integration environment



In the above setting, the multitude of message formats makes the manual processing of the huge amount of information received by the user a very challenging task, especially when rapid business decisions are required (for example warning a pilot about sudden change of circumstances in his flight's path).

Figure 2: Visual representation of temporal and spatial messages in the proposed integrated environment



This is especially true when there is a temporal and/or a spatial overlapping of information between the received messages, or when the messages are numerous. In the above example, the pilot would not be interested in any temporal changes outside his flight's time interval, or any spatial changes outside his flight's route channel – the optimum information would be a set of data representing as accurately as possible the new predicted conditions for his flight.

Our proposal (shown visually in Figure 2) is to integrate the aviation reference data with the other types of temporal and spatial messages (e.g. weather data). This is done by creating an integrated repository in XML format, which provides a synchronised view of the temporal and spatial validity of the received messages, and it makes possible to automate the 'time' and 'space' filtering of the data.

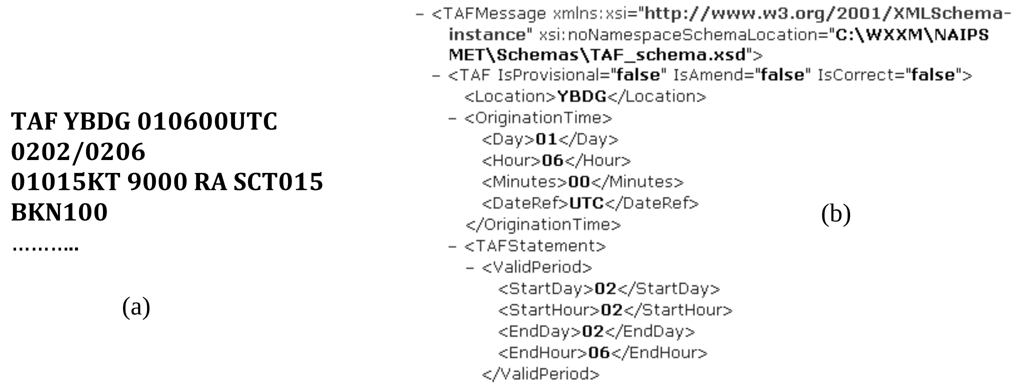
1.1. Temporal features in XML aviation data

The temporal aspect in XML documents is usually represented as date and/or time elements, and it can have at least three dimensions:

- *Transactional date/time* – that is, the date/time when the information is created;
- *Validity interval* – that is, the date/time range when the information is relevant;
- *Usage date/time* – that is, the date/time when the information is consumed by the end user.

To exemplify these time dimensions, Figure 3 shows an example of a TAF (Terminal Area Forecast) message where the *transactional time* is 01/03/2010 while the *validity interval* of the message is between 2am and 6am on 02/03/2010. In Figure 3 we show both a partial regular (text) TAF version, as well as its XML representation.

Figure 3: Example of an XML weather document with data having different transactional and validity times



TAF YBDG 010600UTC
0202/0206
01015KT 9000 RA SCT015
BKN100
………..

(a)

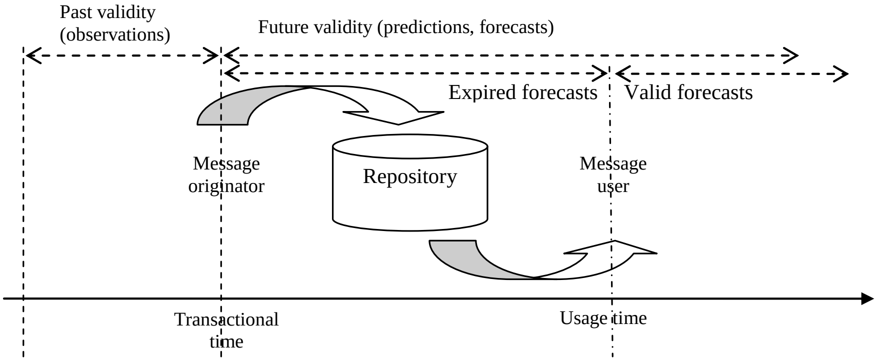
```
- <TAFMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xsi:noNamespaceSchemaLocation="C:\WXXM\NAIPS
    MET\Schemas\TAF_schema.xsd">
  - <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
      <Location>YBDG</Location>
    - <OriginationTime>
        <Day>01</Day>
        <Hour>06</Hour>
        <Minutes>00</Minutes>
        <DateRef>UTC</DateRef>
      </OriginationTime>
    - <TAFStatement>
      - <ValidPeriod>
          <StartDay>02</StartDay>
          <StartHour>02</StartHour>
          <EndDay>02</EndDay>
          <EndHour>06</EndHour>
        </ValidPeriod>
```

(b)

In an environment with a high dynamicity of data (such as weather data which streams continuously from the Bureau of Meteorology (BOM) towards ASA, or the NOTAM (Notice to Airmen) messages created daily in Australia), the interval between the transactional time (at the origin) and the usage time (at the consumer) could be very short. Some types of aviation messages received by ASA (e.g. NOTAM) would not have a past validity compared to the transactional date/time, but always a current or future validity.

Moreover, a characteristic of some aviation data is the flexibility of validity date/time for the messages. For weather data for example, the validity date/time can be prior to the transactional time (e.g. for weather observations), or after the transactional time (e.g. for weather forecasts). However, it depends on the actual usage time whether a certain piece of information represents a forecast *for the user*: Messages with validity interval after the usage time could be described as 'valid forecasts', whereas messages with validity interval prior to the usage time could be 'expired forecasts'. A visual representation of different validity intervals is shown in Figure 4.

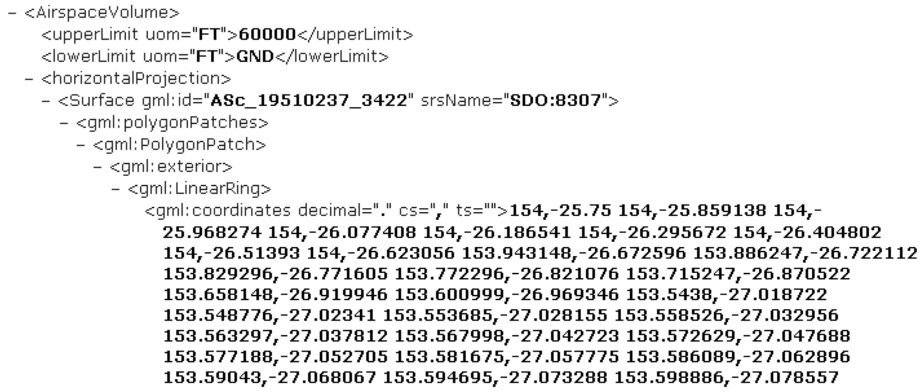
Figure 4: Visual representation of flexible temporal feature in aviation data



1.2. Spatial features in XML aviation data

The spatial aspect in the XML standards is usually represented as elements described in a GML-compliant manner (Geography Markup Language (GML), 2010). For example, a <Location> element would be described as a GML Point, whereas an <Area> or <Surface> element would be described as a GML Surface, GML LinearRing etc. In other words, the spatial elements describe the spatial validity of the XML document based on the schema. Figure 5 shows an example of XML document where an airspace volume location is given in a GML LinearRing representation. Note that Figure 5 shows only a part of the sequence of coordinates which gives the horizontal projection of the airspace.

Figure 5: Example of GML representation of an airspace horizontal projection

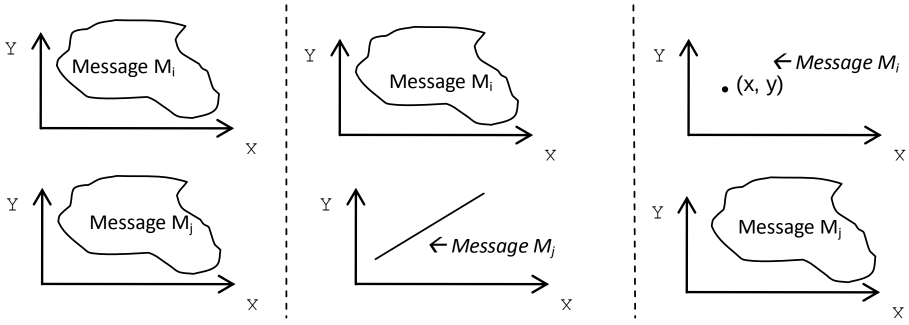


```
- <AirspaceVolume>
    <upperLimit uom="FT">60000</upperLimit>
    <lowerLimit uom="FT">GND</lowerLimit>
  - <horizontalProjection>
    - <Surface gml:id="ASc_19510237_3422" srsName="SDO:8307">
      - <gml:polygonPatches>
        - <gml:PolygonPatch>
          - <gml:exterior>
            - <gml:LinearRing>
                <gml:coordinates decimal="." cs="," ts="">154,-25.75 154,-25.859138 154,-
                  25.968274 154,-26.077408 154,-26.186541 154,-26.295672 154,-26.404802
                  154,-26.51393 154,-26.623056 153.943148,-26.672596 153.886247,-26.722112
                  153.829296,-26.771605 153.772296,-26.821076 153.715247,-26.870522
                  153.658148,-26.919946 153.600999,-26.969346 153.5438,-27.018722
                  153.548776,-27.02341 153.553685,-27.028155 153.558526,-27.032956
                  153.563297,-27.037812 153.567998,-27.042723 153.572629,-27.047688
                  153.577188,-27.052705 153.581675,-27.057775 153.586089,-27.062896
                  153.59043,-27.068067 153.594695,-27.073288 153.598886,-27.078557
```

As shown in Figure 5, GML uses pairs of coordinates (latitude, longitude) to represent points, lines, shapes etc. This makes it virtually impossible for an end user to do spatial computation manually on the fly to determine the spatial validity of the resulting message; hence, a system which allows automated integration and querying of spatial information is needed. When two or more messages have different spatial validity, it becomes increasingly critical to be able to calculate the identical, overlapping or complementing spatial data validity of the resulting integrated message. See Figure 6 for few visual examples of identical and overlapping spatial validity of two messages M_i and M_j.

A real life example for the identical spatial validity in Figure 6 could be two messages referring to an airport, where one is a NOTAM advising the closure of a runway, and the other is a TAF predicting thick fog above the airport. An example of non-identical spatial validity could be when one message predicts heavy storms in a particular airspace: in this case the routes which traverse this airspace have their own spatial validities, which overlap with the spatial validity of the storm-predicting message.

Figure 6: Visual scenarios of identical and overlapping spatial validity



(a) Identical spatial validity

(b) two cases of overlapping spatial validity

1.3. Scope of the paper

At a scientific level, the problem of integrating reference aviation data with spatial and temporal weather information (or any other type of transportation data which has spatial and temporal features, for this matter) translates into the necessity to integrate two or more different standards in which data is currently represented and exchanged. Because many aviation data domains make use nowadays of XML to represent their standards, it means that the end goal is the ability to integrate heterogeneous XML schemas having spatial and temporal features.

As shown in Section 2 (Related Work), the aspect of integrating generic heterogeneous XML schemas has been thoroughly researched; hence, it does not make the object of the present paper. However, very little research work has been done to look at the aspect of integrating XML schemas with temporal and spatial features. It is therefore the object of this paper to propose a model to deal specifically with the integration of those XML schemas which have temporal and spatial features, with direct applicability in the transportation industry.

The rest of the paper is organised as follows: Section 2 describes the existing related work in the area of XML schema integration; Section 3 proposes our model to deal with temporal and spatial schema integration. Section 4 gives a case study based on two XML-based standards, and finally Section 5 concludes the paper.

2. Related work

Research work in the area of XML schema integration has started to become a priority only in the last decade (Christophides 2000, Passi et al 2002, Sakamuri et al.2003). This has been triggered by large amounts of data stored in heterogeneous databases, partly web data, which had to be integrated for the use of the decision support systems.

Generally, the integration of XML schemas is proposed to be done in a layered approach. In (Passi et al 2002) and (Sakamuri et al.2003) the authors propose to have a *pre-integration* phase, where the elements, attributes and data types are extracted from the initial XML schemas. This is followed by a *comparison* phase where any conflicts or correspondence between elements are identified, and finally an *integration* (or *merging*) phase where the elements and attributes are merged, and the global (integrated) XML schema is produced. To query the data, the user needs access to the global schema, obtained from the integration of the individual schemas. The queries on the global schema are transformed in local queries, by using the mapping between the initial elements and the global elements and attributes provided during schema model phase (Sakamuri et al.2003).

To the best of our knowledge, most of the existing approaches are dealing with integration of similar concepts or attributes, rather than relevance in terms of time and space. In our opinion, integration of temporal and spatial features is critical because they describe the validity of the XML documents data.

During the comparison phase in the existing generic approaches, it is not sufficient to identify and solve an eventual conflict of names for the temporal/spatial elements or attributes, but a specific method to deal with the temporal/spatial validity of the data is mandatory. The next section presents our proposed technique to solve this problem.

3. Proposed approach for temporal and spatial integration

This section discusses the problem and proposes the model to deal with temporal and spatial XML schema integration, respectively.

3.1. Integrating temporal features – problem identification

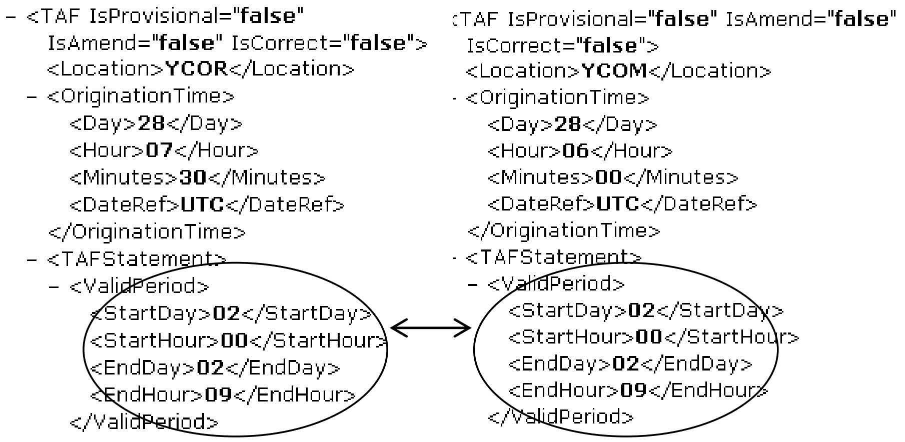
As shown in Figure 3, temporal information in XML documents can often appear as two elements (or attributes) which form a date range (e.g. <StartDay>, <EndDay>, <ValidFrom>, <ValidTo>), or as a single element or attribute, when the data represents a specific point in time (e.g. <OriginationTime>, <CreateDate> etc).

As described earlier in Section 1, different messages could have different periods of validity, depending on the application which creates them. We have identified two scenarios which exist, no matter the exact representation of the temporal elements in each schema.

Definition 1 Given two XML documents D_1 and D_2, where the temporal validity is represented by two sets of nodes values, TN_1 and TN_2 respectively, then document D_1 has *identical temporal validity* with D_2 if and only if $TN_1 = TN_2$ ($\forall$ N $\in TN_1$ then N $\in TN_2$, and $\forall$ N $\in TN_2$ then N $\in TN_1$).

Figure 7 shows an example where the origination time and the location targeted are different, but the temporal validity for the two TAF (weather) messages is identical.

Figure 7: Identical temporal validity for two weather (TAF) messages



```
- <TAF IsProvisional="false"
    IsAmend="false" IsCorrect="false">
    <Location>YCOR</Location>
  - <OriginationTime>
      <Day>28</Day>
      <Hour>07</Hour>
      <Minutes>30</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>02</StartDay>
        <StartHour>00</StartHour>
        <EndDay>02</EndDay>
        <EndHour>09</EndHour>
      </ValidPeriod>
```

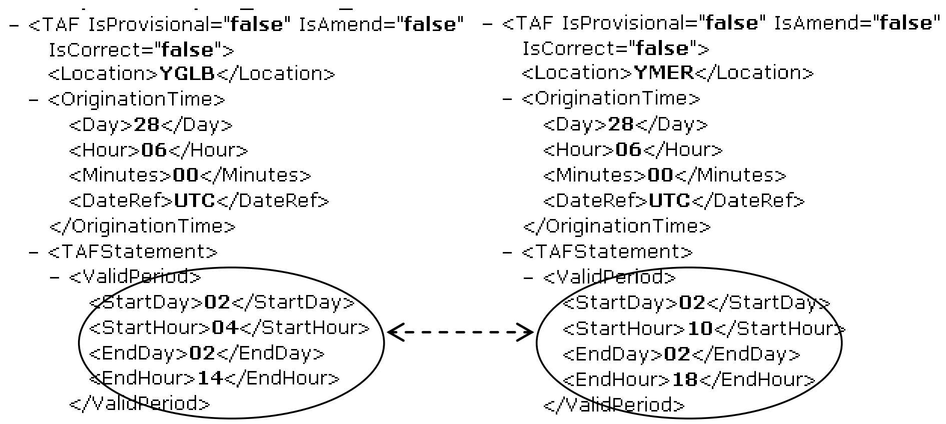
```
:TAF IsProvisional="false" IsAmend="false"
 IsCorrect="false">
 <Location>YCOM</Location>
· <OriginationTime>
   <Day>28</Day>
   <Hour>06</Hour>
   <Minutes>00</Minutes>
   <DateRef>UTC</DateRef>
 </OriginationTime>
· <TAFStatement>
 - <ValidPeriod>
     <StartDay>02</StartDay>
     <StartHour>00</StartHour>
     <EndDay>02</EndDay>
     <EndHour>09</EndHour>
   </ValidPeriod>
```

Definition 2 Given two XML documents D_1 and D_2, where the temporal validity is represented by two sets of temporal nodes values, TN_1 and TN_2 respectively, then document D_1 has *non-identical temporal validity* with D_2 if:

- Temporal validity is overlapping, that is, $TN_1 \cap TN_2 \neq \varnothing$ and $TN_1 \neq TN_2$, or
- Temporal validity is non-overlapping, that is $TN_1 \cap TN_2 = \varnothing$.

Figure 8 shows an example of overlapping temporal validity. In Figure 8, the message for 'YGLB' location (left) is valid between 4am and 2pm, while the message for 'YMER' location (right) is valid between 10am and 6pm. There is therefore a window of 4 hours (between 10am and 2pm) when both messages are valid.

Figure 8: Overlapping temporal validity for two weather (TAF) messages



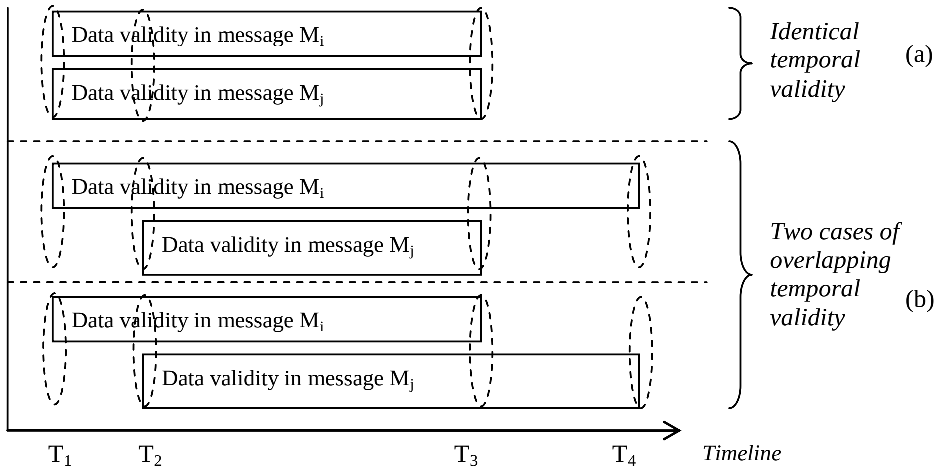
```
- <TAF IsProvisional="false" IsAmend="false"
    IsCorrect="false">
    <Location>YGLB</Location>
  - <OriginationTime>
      <Day>28</Day>
      <Hour>06</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>02</StartDay>
        <StartHour>04</StartHour>
        <EndDay>02</EndDay>
        <EndHour>14</EndHour>
      </ValidPeriod>
```

```
- <TAF IsProvisional="false" IsAmend="false"
    IsCorrect="false">
    <Location>YMER</Location>
  - <OriginationTime>
      <Day>28</Day>
      <Hour>06</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>02</StartDay>
        <StartHour>10</StartHour>
        <EndDay>02</EndDay>
        <EndHour>18</EndHour>
      </ValidPeriod>
```

Naturally, the temporal overlapping is easy to see in this small example, but on a larger scale, when high volume of most likely larger messages are exchanged, the relation between the temporal validities of the critical information might not be as easy to identify.

Figure 9 shows a more generic representation of the temporal scenarios identified in Definitions 1 and 2, and identifies some query problems.

Figure 9: Scenarios of temporal validity in messages (XML documents) with temporal features

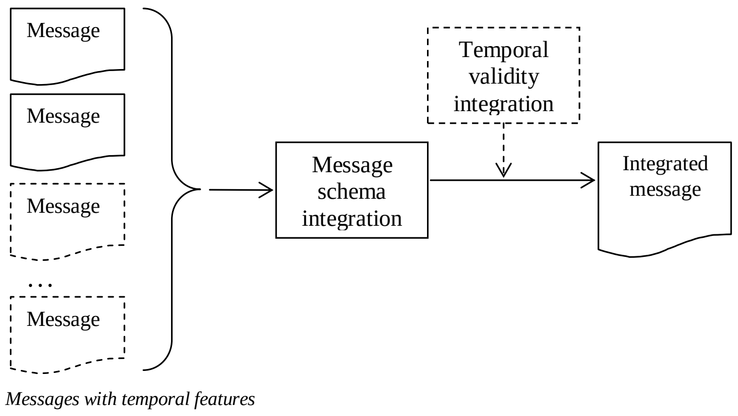

Data validity in message M_i
Data validity in message M_j
Identical temporal validity
(a)
Data validity in message M_i
Data validity in message M_j
Data validity in message M_i
Data validity in message M_j
Two cases of overlapping temporal validity
(b)
T_1
T_2
T_3
T_4
Timeline

In Figure 9(a) data from messages M_i and M_j is valid for the same interval, which is $T_1 – T_3$. Hence, if the user applies a join query on both messages for T_1, T_2, or T_3 he/she will always receive the same result, namely both messages (M_i and M_j).

In Figure 9(b) data from messages M_i and M_j is not valid for the same intervals. In the first case of temporal overlapping, a query for T_1 will only return M_i, a query for T_2 or T_3 will return M_i and M_j, and a query for T_4 will only return M_i again; similar logic applies for the second case of temporal overlapping.

Given the flexibility in expressing temporal validity of data in each message as shown previously (identical or non-identical temporal validity), the integration of temporal features requires a specific approach. The critical question is: how the messages should be integrated, so that the temporal validity of the final integrated message is the correct (intended) one? The problem is visually represented in Figure 10, where we propose to introduce a new step in the integration process, namely the "temporal validity integration", performed after the schema of the integrated message has been defined.

Figure 10: Problem identification in temporal integration of XML messages


Message
Message
Message
…
Message
Messages with temporal features
Message schema integration
Temporal validity integration
Integrated message

3.2. Integrating temporal features – proposed approach

To solve the problem identified in the previous section, we introduce the concept of *temporal validity window*. These windows will be created in the integrated XML document depending on the type of temporal validity in the raw XML messages, as follows:

- ***The initial XML messages have identical temporal validity***

In this case a *single temporal validity window* will be included in the integrated XML message. In other words, the final integrated XML message (document) will have a single set of data, valid for the same (i.e. a single) period of validity.

Definition 3 Given two XML documents D_1 and D_2, which are in an identical temporal validity relationship (TN_1=TN_2, where TN_1 and TN_2 are two sets of nodes giving the temporal validity for documents D_1 and $D_{2,}$ respectively), then the integrated XML document D = $D_1 \cup D_2$ will have the a *single temporal validity window* TVW= TN_1=TN_2.

- ***The initial XML documents have non-identical temporal validity***

In this case, *more than one temporal validity window* will be created in the integrated XML message, based on the individual temporal validity periods of the initial XML messages. In other words, the integrated XML message (document) will contain multiple sets of data, valid for multiple periods of validity.

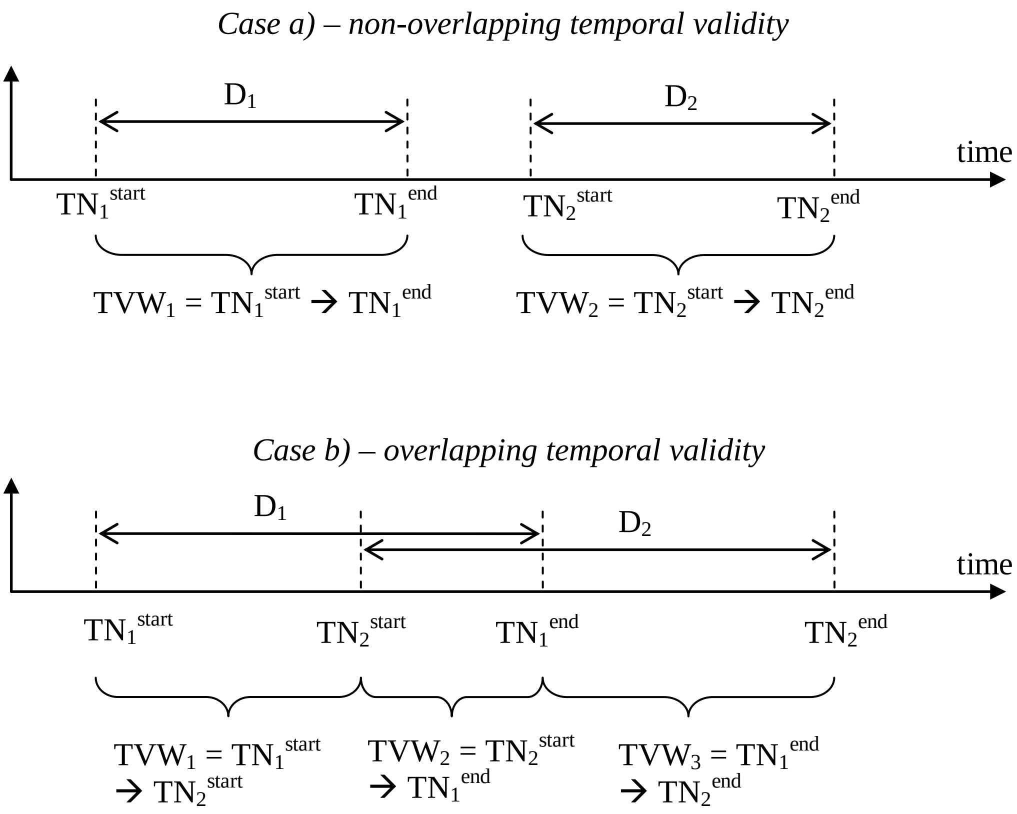
Definition 4 Given two XML documents D_1 and D_2, which are in a non-identical temporal validity relationship ($TN_1 \neq TN_{2,}$ where TN_1 and TN_2 are two sets of nodes giving the temporal validity for documents D_1 and $D_{2,}$ respectively), then the integrated XML document D = $D_1 \cup D_2$ will have a *temporal validity window for each overlapping and non-overlapping segments* of TN_1 and $TN_{2.}$

Example: Given TN_1^{start}, TN_2^{start}, TN_1^{end} and TN_2^{end} respectively the start and end of each temporality interval (TN_1 = {TN_1^{start} → TN_1^{end} }, TN_2 = { TN_2^{start} → TN_2^{end}}), then the integrated document D = $D_1 \cup D_2$ will have temporal validity windows as follows:

- Two temporal validity windows if there was a non-overlapping temporality of D_1 and D_2 ($TN_1 \cap TN_2 = \emptyset$):
  TVW_1 = {TN_1^{start} → TN_1^{end} }, TVW_2 = {TN_2^{start} → TN_2^{end} }
- Three temporal validity windows if there was an overlapping temporality of D_1 and D_2 ($TN_1 \cap TN_2 \neq \emptyset$ and $TN_1 \neq TN_2$), e.g :
  TVW_1 = {TN_1^{start} → TN_2^{start}}, TVW_2 = {TN_2^{start} → TN_1^{end}}, TVW_3 = {TN_1^{end} → TN_2^{end}}

Figure 11 gives a visual representation of the concepts introduced by Definition 4. Figure 11(a) shows the case of non-overlapping temporality of D_1 and D_2, whereas Figure 11(b) shows a case of overlapping temporality of D_1 and D_2 (note that other overlapping scenarios might exist).

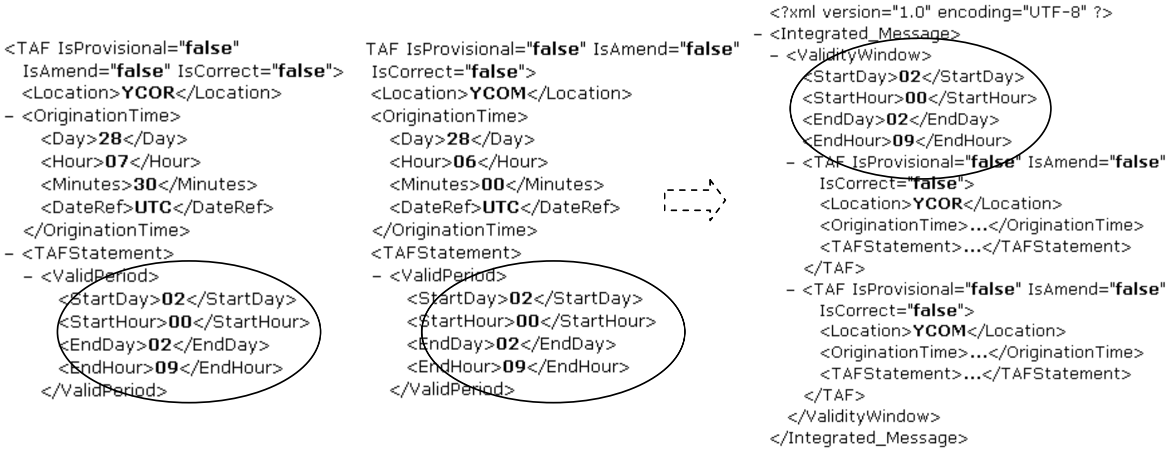
Figure 11: Visual representation of multiple temporal validity windows


Case a) – non-overlapping temporal validity
D_1
D_2
time
TN_1^{start}
TN_1^{end}
TN_2^{start}
TN_2^{end}
$TVW_1 = TN_1^{start} \rightarrow TN_1^{end}$
$TVW_2 = TN_2^{start} \rightarrow TN_2^{end}$
Case b) – overlapping temporal validity
D_1
D_2
time
TN_1^{start}
TN_2^{start}
TN_1^{end}
TN_2^{end}
$TVW_1 = TN_1^{start} \rightarrow TN_2^{start}$
$TVW_2 = TN_2^{start} \rightarrow TN_1^{end}$
$TVW_3 = TN_1^{end} \rightarrow TN_2^{end}$

To exemplify the approach, see Figure 12 for a case of identical temporal validity (which integrates the XML messages from Figure 7) and Figure 13 for a case of non-identical overlapping temporal validity (which integrates the XML messages from Figure 8).

In Figure 12, the final integrated document has a single validity window between midnight and 9am, and then the content of each individual TAF message is listed for each location.

Figure 12: Example of temporal integration of XML messages with identical temporal validity



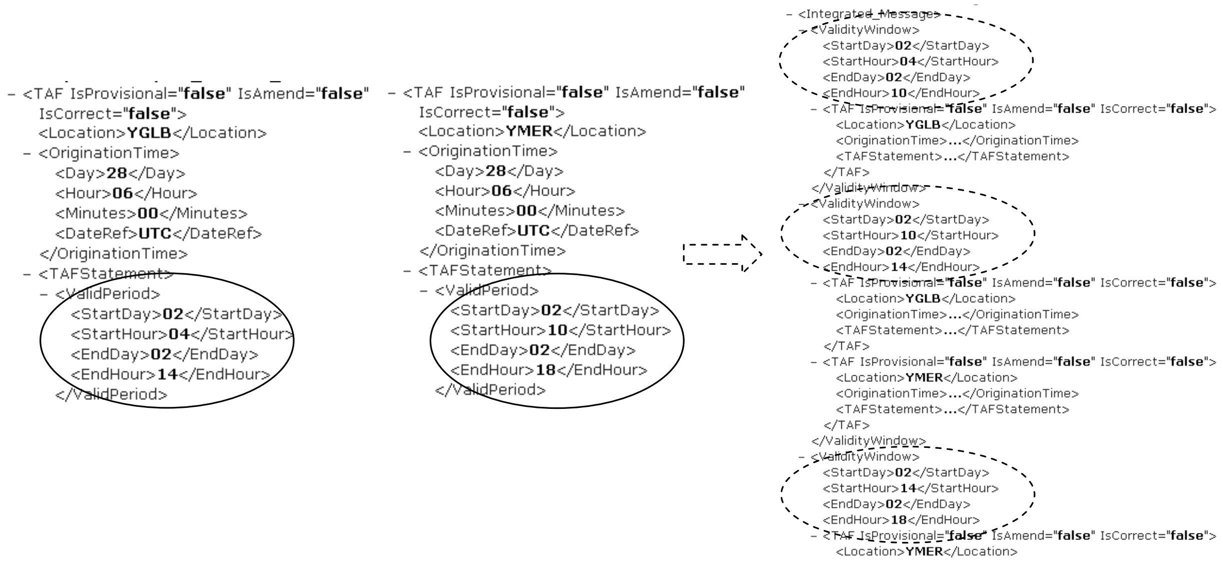
```
<TAF IsProvisional="false"
  IsAmend="false" IsCorrect="false">
  <Location>YCOR</Location>
- <OriginationTime>
    <Day>28</Day>
    <Hour>07</Hour>
    <Minutes>30</Minutes>
    <DateRef>UTC</DateRef>
  </OriginationTime>
- <TAFStatement>
  - <ValidPeriod>
      <StartDay>02</StartDay>
      <StartHour>00</StartHour>
      <EndDay>02</EndDay>
      <EndHour>09</EndHour>
    </ValidPeriod>
```

```
TAF IsProvisional="false" IsAmend="false"
 IsCorrect="false">
 <Location>YCOM</Location>
 <OriginationTime>
   <Day>28</Day>
   <Hour>06</Hour>
   <Minutes>00</Minutes>
   <DateRef>UTC</DateRef>
 </OriginationTime>
 <TAFStatement>
 - <ValidPeriod>
     <StartDay>02</StartDay>
     <StartHour>00</StartHour>
     <EndDay>02</EndDay>
     <EndHour>09</EndHour>
   </ValidPeriod>
```

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Integrated_Message>
  - <ValidityWindow>
      <StartDay>02</StartDay>
      <StartHour>00</StartHour>
      <EndDay>02</EndDay>
      <EndHour>09</EndHour>
    - <TAF IsProvisional="false" IsAmend="false"
        IsCorrect="false">
        <Location>YCOR</Location>
        <OriginationTime>...</OriginationTime>
        <TAFStatement>...</TAFStatement>
      </TAF>
    - <TAF IsProvisional="false" IsAmend="false"
        IsCorrect="false">
        <Location>YCOM</Location>
        <OriginationTime>...</OriginationTime>
        <TAFStatement>...</TAFStatement>
      </TAF>
    </ValidityWindow>
  </Integrated_Message>
```

Figure 13 shows an example of non-identical temporal validity, whereas the final integrated document has multiple validity windows and for each window only the applicable TAF content is listed.

Figure 13: Example of temporal integration of XML messages for overlapping temporal validity



```
- <TAF IsProvisional="false" IsAmend="false"
    IsCorrect="false">
    <Location>YGLB</Location>
  - <OriginationTime>
      <Day>28</Day>
      <Hour>06</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>02</StartDay>
        <StartHour>04</StartHour>
        <EndDay>02</EndDay>
        <EndHour>14</EndHour>
      </ValidPeriod>
```

```
- <TAF IsProvisional="false" IsAmend="false"
    IsCorrect="false">
    <Location>YMER</Location>
  - <OriginationTime>
      <Day>28</Day>
      <Hour>06</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>02</StartDay>
        <StartHour>10</StartHour>
        <EndDay>02</EndDay>
        <EndHour>18</EndHour>
      </ValidPeriod>
```

```
- <Integrated_Message>
  - <ValidityWindow>
      <StartDay>02</StartDay>
      <StartHour>04</StartHour>
      <EndDay>02</EndDay>
      <EndHour>10</EndHour>
    - <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
        <Location>YGLB</Location>
        <OriginationTime>...</OriginationTime>
        <TAFStatement>...</TAFStatement>
      </TAF>
    </ValidityWindow>
  - <ValidityWindow>
      <StartDay>02</StartDay>
      <StartHour>10</StartHour>
      <EndDay>02</EndDay>
      <EndHour>14</EndHour>
    - <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
        <Location>YGLB</Location>
        <OriginationTime>...</OriginationTime>
        <TAFStatement>...</TAFStatement>
      </TAF>
    - <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
        <Location>YMER</Location>
        <OriginationTime>...</OriginationTime>
        <TAFStatement>...</TAFStatement>
      </TAF>
    </ValidityWindow>
  - <ValidityWindow>
      <StartDay>02</StartDay>
      <StartHour>14</StartHour>
      <EndDay>02</EndDay>
      <EndHour>18</EndHour>
    - <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
        <Location>YMER</Location>
```

As it can be noticed, in the case of multiple validity windows only the data pertaining to the specific user query will be extracted and broadcasted when necessary. For example, if the weather message was integrated as in Figure 13 and a pilot flew from YMML to YGLB between 3pm and 4pm (1-hour flight), only the TAF message pertaining to that particular interval (that is, the TAF for location 'YMER') will be extracted and passed to him/her. Currently the same pilot is overloaded with information, because before take-off he/she receives all the latest available TAF messages for the locations/areas of interest, no matter their temporal validity interval. We show in the next subsection that spatial validity of the integrated message can also be filtered, so if the pilot was not interested in location 'YMER' (e.g. outside the flight route channel) the message would not be extracted at all.

3.3. Integrating spatial features – problem identification

As mentioned in Section 1, the *spatial aspect* is represented in the aviation XML standards (such as AIXM or WXXM) as GML objects (GML, 2010). For example, a <Location> element would be described as a 'GML Point', an <Area> or <Surface> element would be described as a 'GML Surface', while a <Trajectory> element might be described as a 'GML Line'. Figure 5 showed an example of XML document where an airspace volume location was given in a GML LinearRing representation.

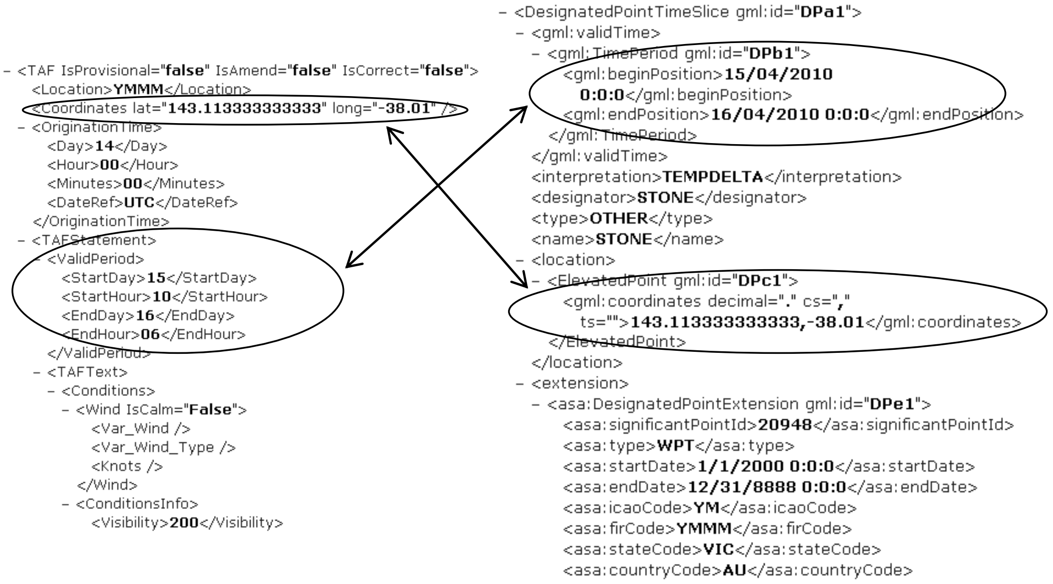
Other standards to represent spatial objects could be used as well, however this is not critical. The essential point is that multiple XML messages can have different spatial validity (such as points, lines, surfaces etc). In the context of the high volume of aviation data exchanged, where each message has its own spatial validity, a technique which allows spatial integration of these messages and their subsequent spatial filtering is therefore vital.

We have identified two scenarios, which exist no matter the exact representation of the spatial features in each schema, i.e. *identical spatial validity* and *non-identical spatial validity*. Their definitions are similar with those for temporal validity (see Section 3.1).

Definition 5 Given two XML document D_1 and D_2, where the spatial validity is represented by two sets of nodes values, SN_1 and SN_2 respectively, then document D_1 has *identical spatial validity* with D_2 if and only if $SN_1=SN_2$ ($\forall$ N $\in$ SN_1 then N $\in$ SN_2 , and $\forall$ N $\in$ SN_2 then N $\in$ SN_1).

Figure 14 shows an example of identical spatial validity. Although the validity interval of the messages is different, they both transmit information valid for the same location (with code 'YMMM').

Figure 14: Example of two XML documents with identical spatial validity



```
- <TAF IsProvisional="false" IsAmend="false" IsCorrect="false">
    <Location>YMMM</Location>
    <Coordinates lat="143.113333333333" long="-38.01" />
  - <OriginationTime>
      <Day>14</Day>
      <Hour>00</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>15</StartDay>
        <StartHour>10</StartHour>
        <EndDay>16</EndDay>
        <EndHour>06</EndHour>
      </ValidPeriod>
    - <TAFText>
      - <Conditions>
        - <Wind IsCalm="False">
            <Var_Wind />
            <Var_Wind_Type />
            <Knots />
          </Wind>
        - <ConditionsInfo>
            <Visibility>200</Visibility>
```

(a)

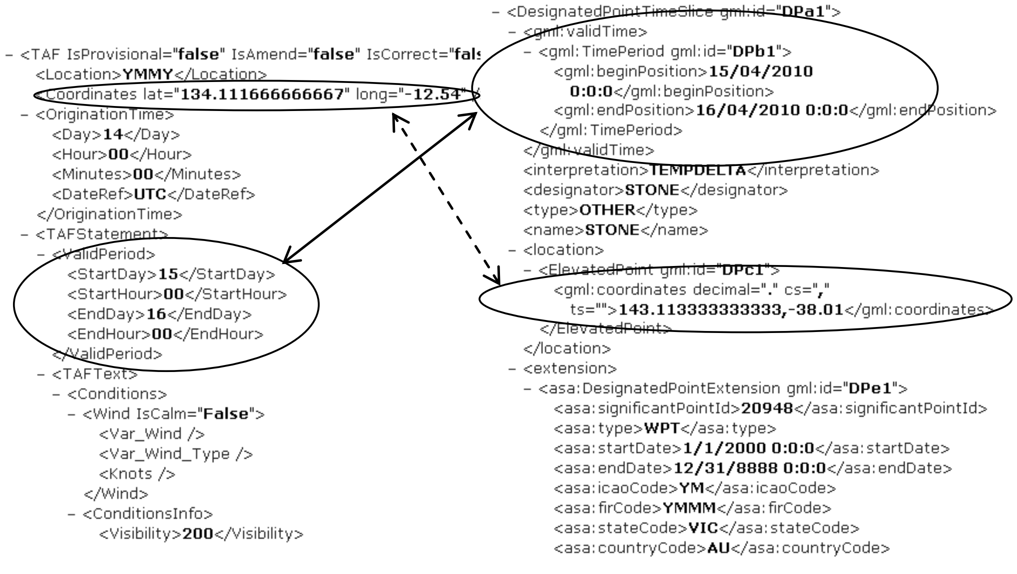
```
- <DesignatedPointTimeSlice gml:id="DPa1">
  - <gml:validTime>
    - <gml:TimePeriod gml:id="DPb1">
        <gml:beginPosition>15/04/2010
          0:0:0</gml:beginPosition>
        <gml:endPosition>16/04/2010 0:0:0</gml:endPosition>
      </gml:TimePeriod>
    </gml:validTime>
    <interpretation>TEMPDELTA</interpretation>
    <designator>STONE</designator>
    <type>OTHER</type>
    <name>STONE</name>
  - <location>
    - <ElevatedPoint gml:id="DPc1">
        <gml:coordinates decimal="." cs=","
          ts="">143.113333333333,-38.01</gml:coordinates>
      </ElevatedPoint>
    </location>
  - <extension>
    - <asa:DesignatedPointExtension gml:id="DPe1">
        <asa:significantPointId>20948</asa:significantPointId>
        <asa:type>WPT</asa:type>
        <asa:startDate>1/1/2000 0:0:0</asa:startDate>
        <asa:endDate>12/31/8888 0:0:0</asa:endDate>
        <asa:icaoCode>YM</asa:icaoCode>
        <asa:firCode>YMMM</asa:firCode>
        <asa:stateCode>VIC</asa:stateCode>
        <asa:countryCode>AU</asa:countryCode>
```

(b)

Definition 6 Given two XML document D_1 and D_2, where the spatial validity is represented by two sets of nodes values, SN_1 and SN_2 respectively, then document D_1 has *non-identical spatial validity* with D_2 if $SN_1 \neq SN_2$ ($\exists$ N $\in$ SN_1 where N $\notin$ $SN_{2,}$ or $\exists$ N $\in$ SN_2 where N $\notin$ SN_1).

Figure 15 shows an example of non-identical spatial validity, where the spatial validities of the documents differ, although they are both valid for the same date/time interval.

Figure 15: Examples of two XML documents, with non-identical spatial validity



```
- <TAF IsProvisional="false" IsAmend="false" IsCorrect="fals
    <Location>YMMY</Location>
    <Coordinates lat="134.111666666667" long="-12.54"
  - <OriginationTime>
      <Day>14</Day>
      <Hour>00</Hour>
      <Minutes>00</Minutes>
      <DateRef>UTC</DateRef>
    </OriginationTime>
  - <TAFStatement>
    - <ValidPeriod>
        <StartDay>15</StartDay>
        <StartHour>00</StartHour>
        <EndDay>16</EndDay>
        <EndHour>00</EndHour>
      </ValidPeriod>
    - <TAFText>
      - <Conditions>
        - <Wind IsCalm="False">
            <Var_Wind />
            <Var_Wind_Type />
            <Knots />
          </Wind>
        - <ConditionsInfo>
            <Visibility>200</Visibility>
```

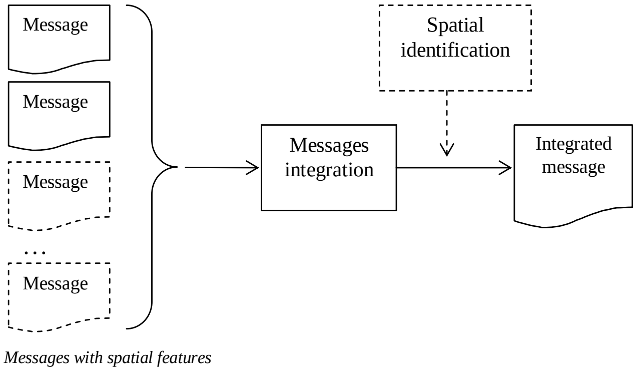
(a)

```
- <DesignatedPointTimeSlice gml:id="DPa1">
  - <gml:validTime>
    - <gml:TimePeriod gml:id="DPb1">
        <gml:beginPosition>15/04/2010
          0:0:0</gml:beginPosition>
        <gml:endPosition>16/04/2010 0:0:0</gml:endPosition>
      </gml:TimePeriod>
    </gml:validTime>
    <interpretation>TEMPDELTA</interpretation>
    <designator>STONE</designator>
    <type>OTHER</type>
    <name>STONE</name>
  - <location>
    - <ElevatedPoint gml:id="DPc1">
        <gml:coordinates decimal="." cs=","
          ts="">143.113333333333,-38.01</gml:coordinates>
      </ElevatedPoint>
    </location>
  - <extension>
    - <asa:DesignatedPointExtension gml:id="DPe1">
        <asa:significantPointId>20948</asa:significantPointId>
        <asa:type>WPT</asa:type>
        <asa:startDate>1/1/2000 0:0:0</asa:startDate>
        <asa:endDate>12/31/8888 0:0:0</asa:endDate>
        <asa:icaoCode>YM</asa:icaoCode>
        <asa:firCode>YMMM</asa:firCode>
        <asa:stateCode>VIC</asa:stateCode>
        <asa:countryCode>AU</asa:countryCode>
```

(b)

Given the flexibility in expressing spatial validity of data in XML messages, the question is how the messages should be integrated, so that the spatial validity of the final integrated message is the correct one. The problem is visually represented in Figure 16, and we propose to introduce a new phase, after the global XML schema has been created (Sakamuri et al.2003), namely the 'spatial identification' phase. The purpose of this new phase is to determine and flag the spatial objects' characteristics in the integrated XML document.

Figure 16: Problem identification in spatial integration of XML messages



3.4. Integration spatial features – proposed approach

As mentioned above, the proposed approach to solve the spatial integration issue is to have a new phase in the integration process, namely *spatial identification*, which will determine and include in the integrated XML document details about spatial objects in the initial XML messages. The algorithm in Figure 17 shows the steps of the proposed spatial identification phase.

Figure 17: Proposed algorithm for spatial identification

```
Algorithm: Spatial_Identification
Input: XML documents/messages $M_i$ and $M_j$ , M integrated XML document without spatial
identification, F={$f_i$,1≤i≤n}, where $f_i$=spatial functions of interest (intersect, distance etc)
Output: M integrated XML document with spatial identifications
1. For each $SpOb_i$ ← spatial object in message $M_i$
2.    For each $SpOb_j$ ← spatial object in message $M_j$
3.        $SpOb_i$_ID ← unique identifier of spatial object $SpOb_i$ in M
4.        $SpOb_j$_ID ← unique identifier of spatial object $SpOb_j$ in M
5.        /* determine type of spatial object in $M_i$
6.        If type($SpOb_i$)=GML_Point then
7.           sType$M_i$='Point'
8.        ElseIf type($SpOb_i$)=GML_LineString then
9.           sType$M_i$='Line'
10.        ElseIf type($SpOb_i$)=GML_Surface then
11.           sType$M_i$='Surface'
12.        Else
13.           sType$M_i$='Other spatial'
14.       End If
15.       /* repeat same logic as in 6-14 above for the type of spatial object in $M_j$ → sType$M_j$
16.       /* add info to document M
17.       M ← <Spatial Type=sType$M_i$ ID=$SpOb_i$_ID>$SpOb_i$</Spatial>
18.       M ← <Spatial Type=sType$M_j$ ID=$SpOb_j$_ID>$SpOb_j$</Spatial>
19.       /* determine applicability of functions $f_i$ on $SpOb_i$ and $SpOb_j$
20.       If $SpOb_i$ spatial_function $SpOb_j$ then
21.           /* add spatial functions info to document M
22.           M ← <Spatial_FunctionName><Spatial>$SpOb_i$_ID</Spatial>
23.                   <Spatial>$SpOb_j$_ID</Spatial></Spatial_FunctionName>
24.       End If
25.    Next $SpOb_j$
26. Next $SpOb_i$
27. Return M integrated XML document with spatial identification
```

Mainly, we propose to identify each spatial object from the initial XML messages in the integrated XML message, and to determine the applicability of any spatial function of interest. For example, if two spatial objects overlap, then the integrated document will flag the intersection of those objects. Note that the algorithm in Figure 17 refers to the GML representation of spatial objects in aviation data, but the same logic applies for other spatial representations, if need be. By flagging the spatial identify/overlapping of pair of incoming messages, this will allow for easy spatial filtering on the integrated document, since the user queries normally would want to know about spatial intersections, or proximity between objects etc. An example of a user query on the spatial integrated data might be "*retrieve all airports which are in the same geographical area as the predicted storm*".

In this section, we have presented our proposed approach to deal with temporal and spatial features, separately, as part of the XML schema integration process. Most of the time the XML messages would contains both temporal and spatial features. In this case, a combination of the both proposed approaches is required.

The next section shows a case study where three types of aviation messages represented as XML messages have both temporal and spatial features. Several examples of user queries on the integrated data are presented, together with a visual representation of the query result, for each of them.

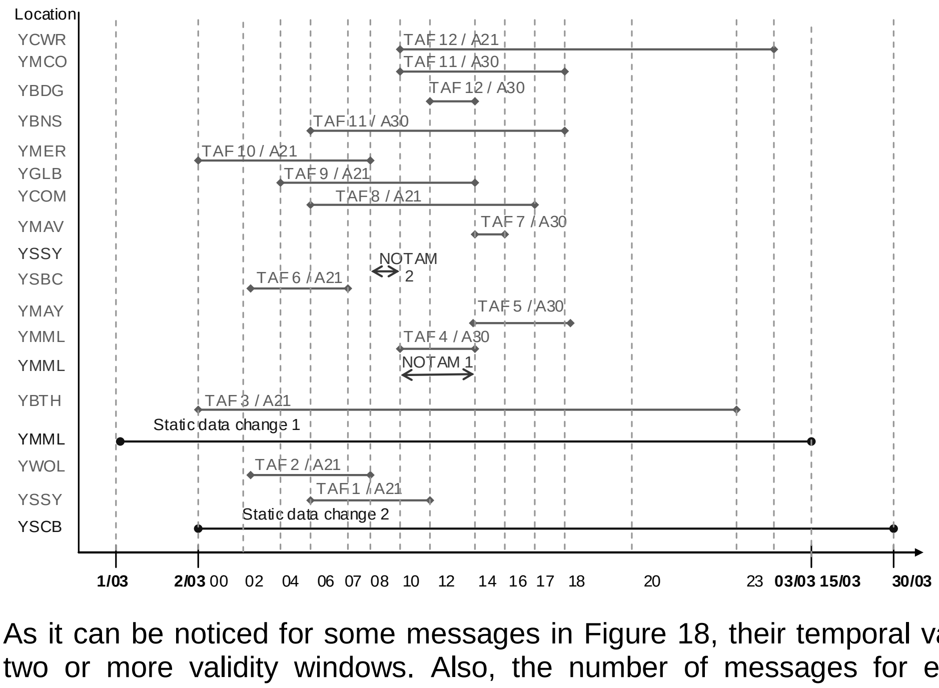
4. A Case Study

Three types of aviation data, with temporal and spatial features, have been collected, integrated and stored in the database using the techniques proposed in Section 3, as follows:

- *2 static data changes received on 01 March 2010*
    - Length of runway 16/34 at Melbourne Airport changed to 3200m between 01 and 15 March;
    - Take-off distance for runway 17/35 at Canberra Airport changed between 02 and 30 March 2010.
- *20 TAF (weather) messages received on 02 March 2010*
    - for Area 30 – 10 TAFs, including for Melbourne;
    - for Area 21 –10 TAFs, including for Sydney and Canberra;
- *2 NOTAMs (Notice to airman) messages received on 02 March 2010*
    - Runway 09/27 at Melbourne airport suspended 10am and 2pm due to rubber removal;
    - Sydney Airport closed between 8am and 10am due to fog.

Figure 18 shows a visual representation of these messages, where the X axis shows the validity time, and the Y axis shows the location for each message.

Figure 18: Sample messages for the case study



As it can be noticed for some messages in Figure 18, their temporal validities overlap over two or more validity windows. Also, the number of messages for each location differs.

Noticeably, although the number of messages in this case study is small (only 24) it is already difficult to answer a query relating to a particular location and time range just by visual analysis of the data.

Few examples of possible queries on the integrated (temporal and spatial) data are:

None or limited filtering, e.g.:

- Get all messages valid for (anytime) today;

Temporal filtering (restrict time of interest), e.g.:

- Get all messages valid between 12pm and 4pm;
- Get TAFs which predict significant changes between 10am and 8pm;
- Get TAFs and NOTAMs valid between 6am and 6pm;

Spatial filtering – single location focus, e.g.:

- Get NOTAMs/TAFs for Melbourne and/or Sydney airports;

Spatial filtering – area focus:

- Get all messages for Canberra Airport;
- Get all radar messages which intersect routes starting at Melbourne airport;
- Get and broadcast only NOTAMs pertaining to route 123;

Combined tempo-spatial focus (time & area of interest), e.g.:

- Get TAFs and NOTAMs for Melbourne Airport, valid 10am - 2pm;
- Get TAFs which predict changes between 10am and 8pm in area 30;
- Get dynamic of radar data between 6am – 8am for Area 21;

The next 4 subsections will present examples for each of the above types of possible queries. For each query there will be a graphical indication of the filter applied, followed by a visual representation of plotting the query result on Bing Maps.

Example of temporal filtering

Query Q1: Get all messages valid between 10am and 4pm

The dotted shape in Figure 19 highlights the messages affected by Q1 filter (covering three validity windows between 10 and 16), and Figure 20 shows the query result on Bing Maps.

Figure 19: Example of temporal filtering –filter applied

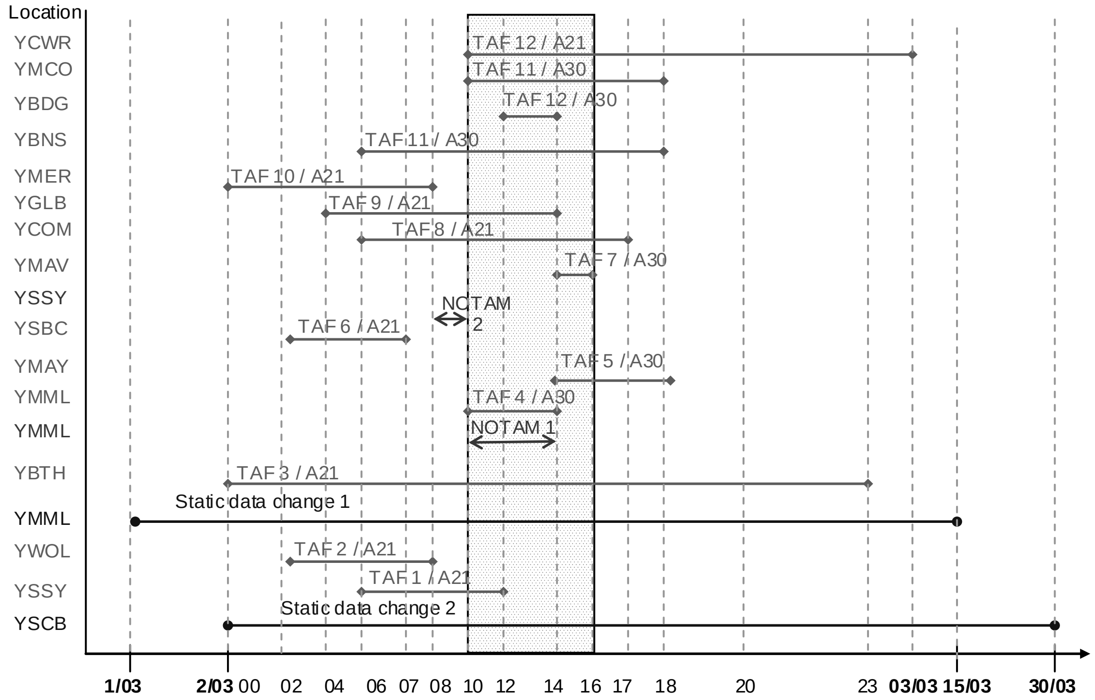
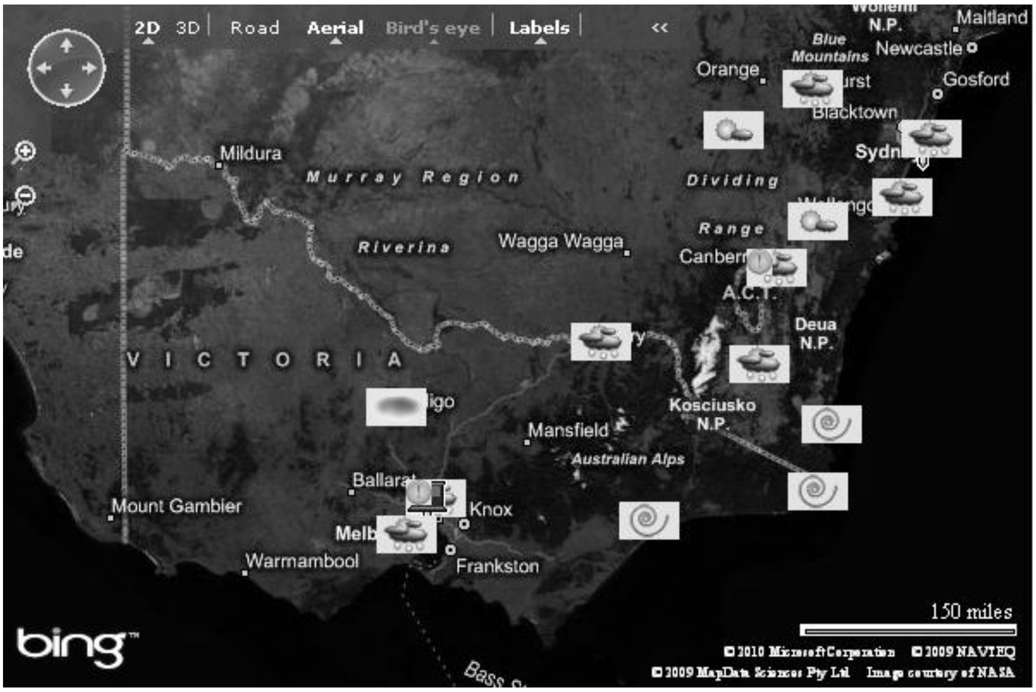


Figure 20: Example of temporal filtering – result



Example of spatial filtering

Query Q2: Get all messages for location YMML

The dotted shapes in Figure 21 highlight the messages affected by Q2 filter (only those where location = YMML) and Figure 22 shows the query result on Bing Maps.

Figure 21: Example of spatial filtering –filter applied

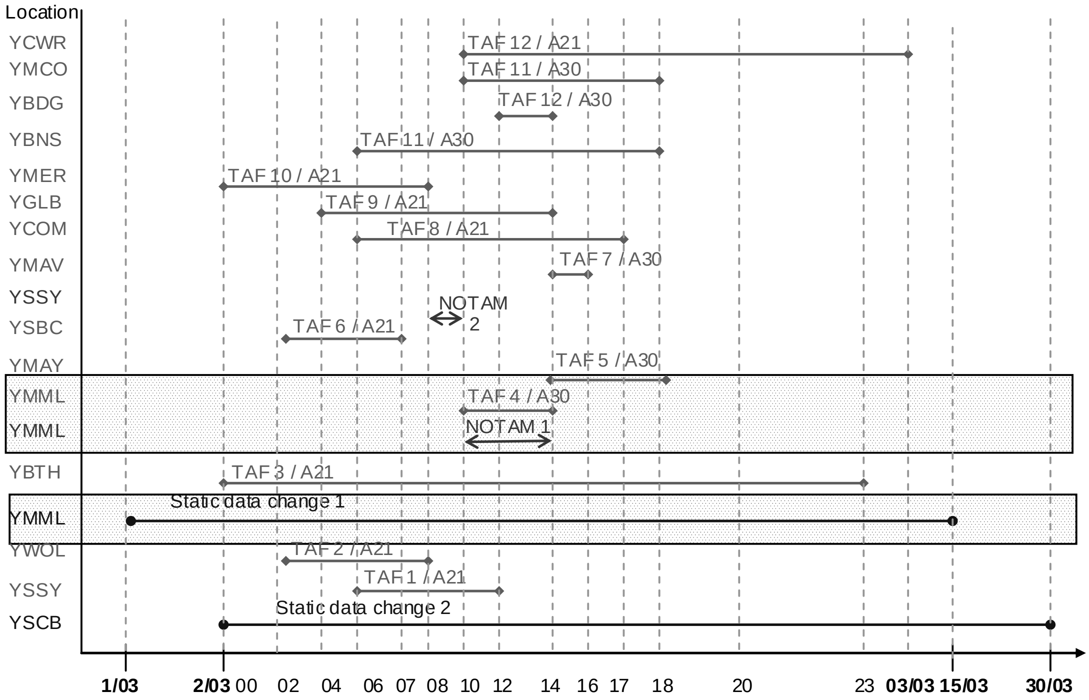
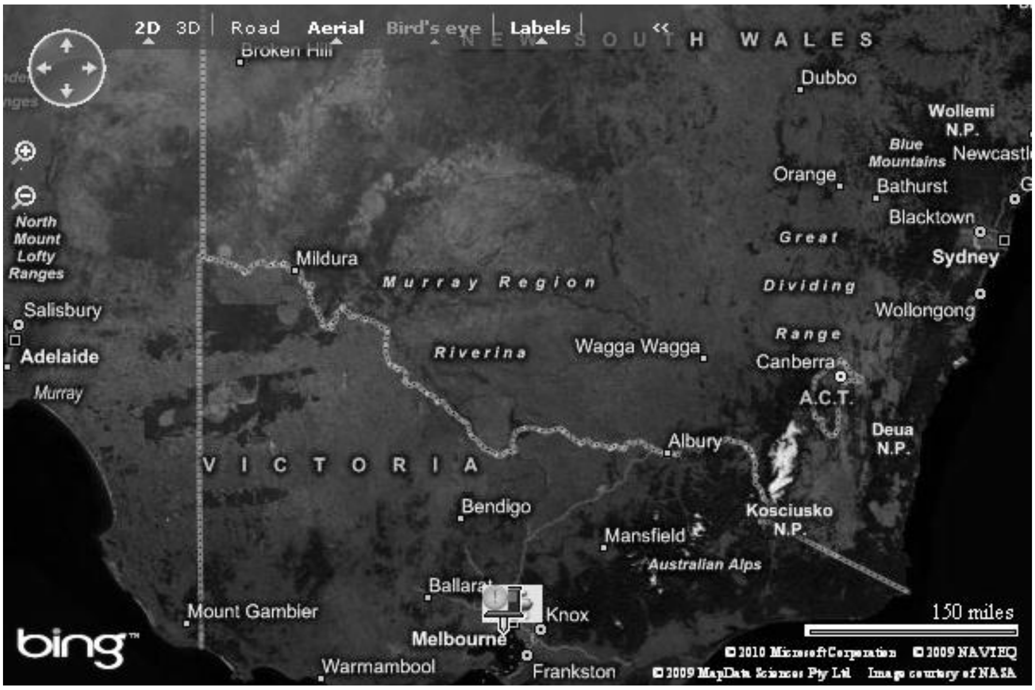


Figure 22: Example of spatial filtering – result



Example of temporal and spatial filtering (I)

Query Q3: Get all messages for Area 30 between 6am and 6pm
The dotted shapes in Figure 23 highlight the messages affected by Q3 filter and Figure 24 shows the result on Bing Maps.

Figure 23: First example of temporal and spatial filtering –filter applied

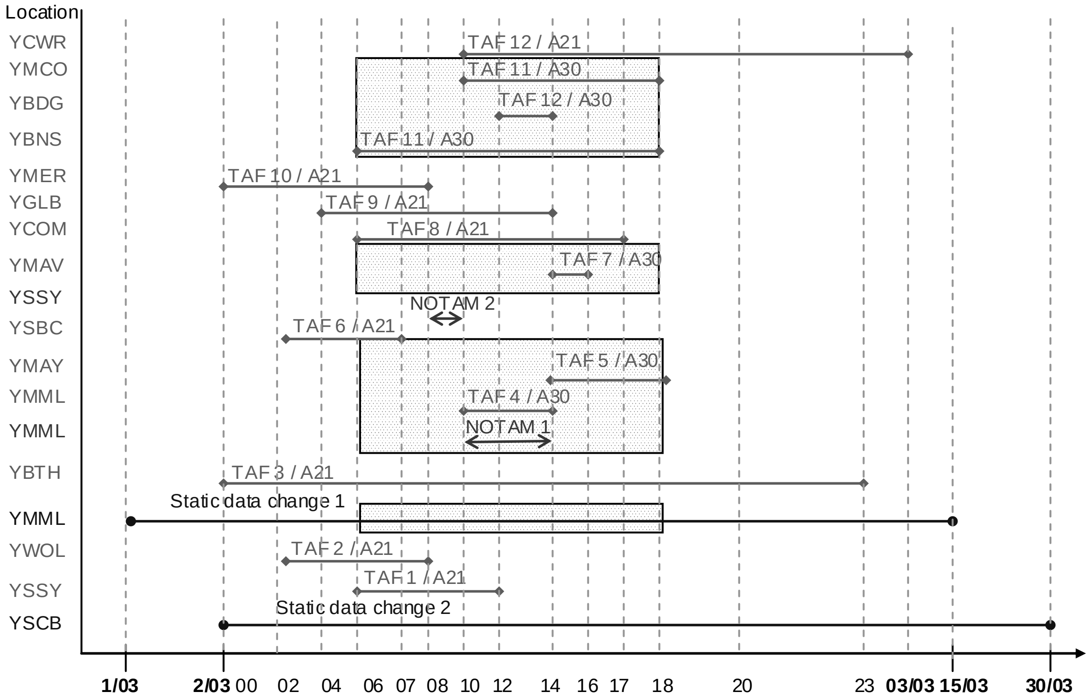
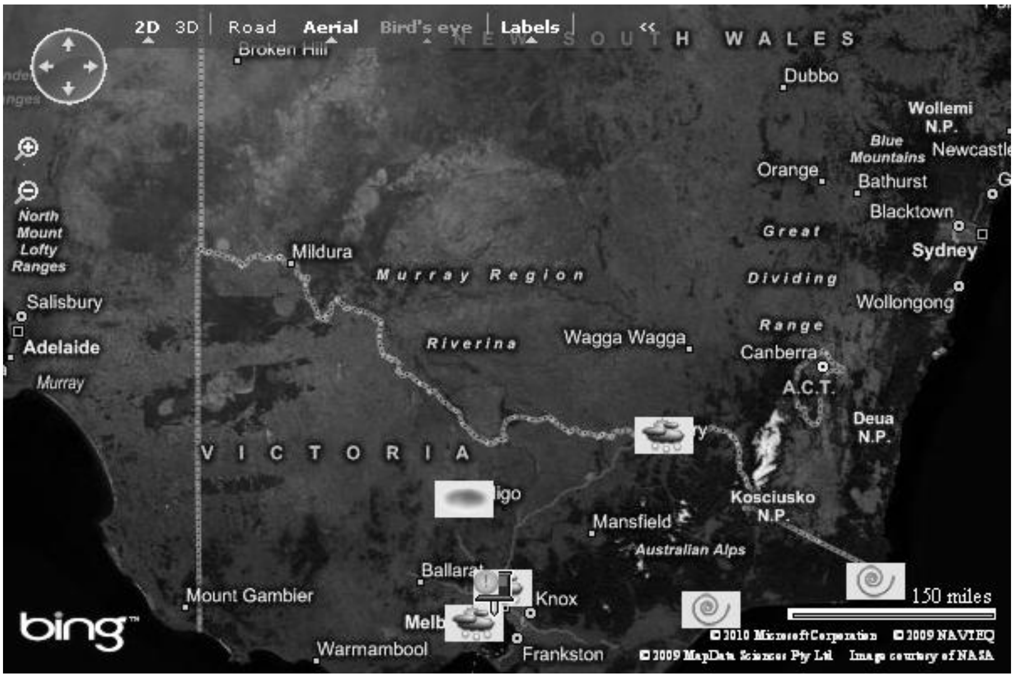


Figure 24: First example of temporal and spatial filtering – result



Example of temporal and spatial filtering (II)

Query Q4: Get all messages for Route 123 (Melbourne to Sydney via Canberra)
The dotted shapes in Figure 25 highlight the messages affected by Q4 filter and Figure 26 shows the result on Bing Maps.

Figure 25: Second example of temporal and spatial filtering – filter applied

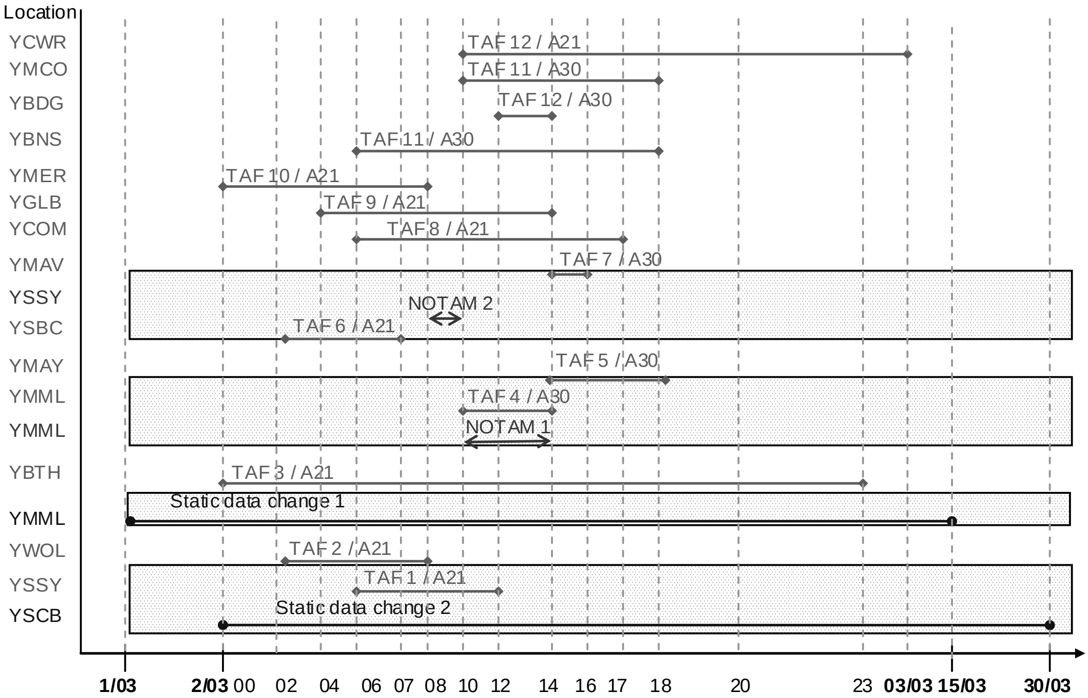
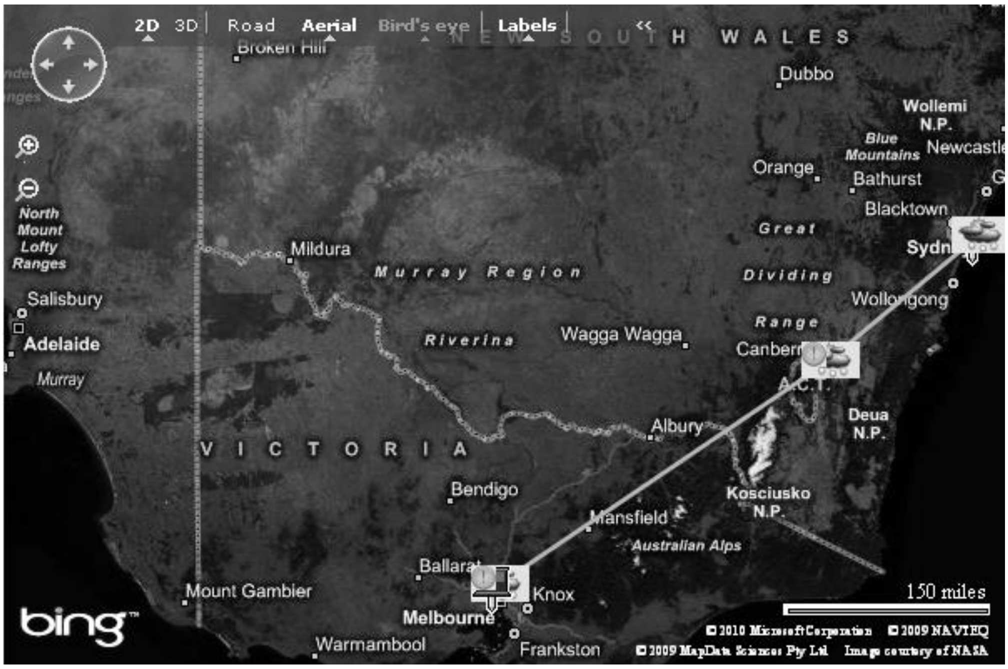


Figure 26: Second example of temporal and spatial filtering – result



5. Conclusions

This paper proposes a method to deal with temporal and spatial features in XML messages. Previous work has looked at the schema-level integration of these messages; however, it has not looked at how their temporal and/or spatial validity are represented in the integrated message. We propose to use the concept of 'temporal validity window' to highlight the validity of different sets of data for different temporal periods, and a 'spatial identification' phase to highlight the spatial relationships between spatial objects in the integrated document. The case study section shows few types of queries on the integrated data, together with a visual representation of the results.

Acknowledgment

This work has been supported by AirServices Australia – La Trobe University industry collaboration grant, 2008-2010.

References:

AIXM schema, online at http://www.aixm.aero/public/standard_page/download.html

WXCM Primer, Aeronautical Information Management, 23 July 2008, online at http://www.eurocontrol.int/aim/public/standard_page/met_wie.html

GML Schema, online at http://schemas.opengis.net/gml/3.1.1/base/, April 2010

LaTrobe University – AirServices Australia, Adoption of AIXM5.0 for ASA, Technical Report, August 2009

Bureau of Meteorology, Review into the Provision of Aerodrome Forecasts (TAF), Information Paper, downloaded August 2009, online at http://www.bom.gov.au/ reguser/by_prod/aviation/

Christophides, V, Cluet, S, Simon, J. On Wrapping Query Languages and Efficient XML Integration, ACM SIGMOD, Dallas, Texas, 2000

Passi, K., Lane, L., Mandria, S.K., Sakamuri, B.C., Mohania, M, and Bhowmick, S.S, A model for XML Schema Integration, EC-Web 2002, LNCS 2455, pp. 193-202, Springer 2002

Sakamuri, B., Mandria, S, Passi, K., Chaudhry, E, Mohania, and Bhowmick, S, AXIS: A XML Schema Integration System, 22nd International Conference on Conceptual Modeling (ER 2003), Lecture Notes in Computer Science 2813, pp. 576-578, Springer 2003