Journal of Universal Computer Science, vol. 16, no. 1 (2010), 22-36
submitted: 1/2/09, accepted: 15/10/09, appeared: 1/1/10 © J.UCS

Integration of Similar Evolving Data Sources for Supporting Decision Making Tasks

Alberto Salguero
(University of Granada, Andalucía, Spain
agsh@ugr.es)

Francisco Araque
(University of Granada, Andalucía, Spain
faraque@ugr.es)

Abstract: Information Systems usually rely on external and independent data sources. When integrating the data to build the integrated repository it is possible to make use of the temporal characteristics of the data sources to improve the whole integration process and the quality of the integrated data, which support the organization's decision-making tasks. In this work the usage of an Ordered Weighted Averaging-based operator is presented as the best option when the data sources refer to similar facts but the data on each data source is expressed with different temporal characteristics. This is a common issue in Information Systems development.

Keywords: data integration, information system, temporal, OWA
Categories: H.2.5, H.2.8, H.4.0

1 Introduction

The ability to integrate data from a wide range of data sources is an important field of research in data engineering. Data integration is a prominent theme in many areas and enables widely distributed, heterogeneous, dynamic collections of information sources to be accessed and handled, supporting, in most cases, the organization's decision-making tasks.

Information Systems (IS) usually rely on external and independent data sources, which the data is extracted from. Due to this fact, finding data sources which refer to measures about the same or similar facts is rather common. The problem arises when these measurements are expressed with different temporal characteristics and need to be integrated into the IS repository. Given a set of data sources, there are basically two issues which should be resolved in order to build an integrated repository:

- identify the concepts in the data source set which refer to the same facts (synonyms, specializations...) and,
- extract and transform each data source value corresponding to each integrated concept so it makes sense integrating all the values which refer to the same concept (scale, level of detail...) and como from the data source set.

This work is focused on solving the latter issue although some considerations have been adopted for facilitating the former. The temporal characteristics of the data sources can be used for improving the quality of the integrated data. In particular, this work focuses on the situations in which the integration of the data should be performed in such a way that the resultant data should be produced more frequently than some of the data sources are. To perform this kind of integration it is assumed that the information to be integrated is related to and varies in a similar way within each of the different data sources. This assumption is true in the case of the IS where detailed information is not usually needed and the underlying information is aggregated for supporting managers' decisions.

The aggregation of the data is carried out by means of the Ordered Weighted Averaging operator (OWA). This operator has been widely applied in decision supporting tasks. Basically, the OWA operator is used when the information given by some members of the group is more representative than the rest but the accuracy of each member's given value varies along time. We can take advantage of this feature of the OWA operator for integrating the data given by some different sources which vary in a similar way. At every instant, the integrated value is set according to the data sources which give more information about the tendency of the measure.

So as to support this task we have developed an architecture which encompasses most of the processes needed to perform the whole integration process. The information about the temporal characteristics of the data source is annotated in the data source scheme by means of an ontology based model. This model has been created for supporting tasks such as the identification of similar concepts in the data sources schemes. It is possible to take advantage of the reasoning capabilities of the ontologies for these kinds of tasks. The temporal metadata in the data sources schemes serve as the input for the integration algorithm presented in this work.

There have been some approaches to the data integration problem. In [Russo 04] a multilayered, fuzzy architecture is presented. Although it is mainly dedicated to image segmentation purposes, their architecture is general enough to support the data integration process in an IS environment. In [Abdulghafour et al. 94] another fuzzy method for aggregating values coming from different sources is presented. They propose a formula for determining the weight of each value. Nevertheless, none of the previous works encompass the aspects of the integration of the data according to its temporal and spatial characteristics. There are other interesting approaches which try to incorporate this information within the integration process and allow the integration of data with different periods of time, also using OWA-based aggregators [Xu 08], [Xu, Yager 08]. The main difference between those methods and the one presented in this paper is the necessity of having the same number of values to be integrated in each data source. In our case, each data source can provide a different number of values for each period of time.

The remaining part of this paper is organized as follows. In the following section a running example is presented in order to ease the understanding of the examples proposed in subsequent sections. In the same section some basic concepts are also introduced and our previously related works are reviewed. In section 3 the integration algorithm is presented. Its results are compared to other similar techniques in section 4. Finally, the concluding remarks are exposed.

2 Preliminaries

In this section some basic concepts necessary for understanding the proposed algorithm are introduced. Firstly, a simple integration example is presented in order to clarify the problem we face in this work. Then, the general integration architecture is reviewed briefly. Some basic temporal concepts are reviewed and, finally, the OWA operator is explained.

2.1 A running example: the temperature of the province of Granada

For our better understanding of the process, in the rest of the work we will make references to the example given in this section.

It is easy to see that the simple average aggregation operator can only be used when the quantity of values provided by each of the data sources implicated are similar. Using the average aggregation operator when there is a data source with a greater number of values, makes the final, integrated result tends to that data source.

This latter situation, which appears somewhat extraordinary, is the situation we had to face when trying to aggregate the temperature of each province in Andalucía (southern region of Spain) for our decision support system for soaring site recommendation [Araque et al. 06]. The information about the temperature for this system is gathered from two mainly kinds of data sources:

- US National Weather Web Service for getting weather hourly measurements (temperature, pressure, humidity, etc) at Granada airport.
- Granada City Council‘s website. It is possible to obtain a temperature value for the city of Granada every half an hour.
- Motril City Council‘s website. The web site of this City Council, in the Granada province, gives only information about the maximum and the minimum daily temperature.

Following the rule about the level of the detail of the intended integrated repository we have set before in this section, we can use a simple arithmetic average operator for integrating the values if we need the monthly temperature or a longer period because we have enough data from each data source. This is not the case if we need to give an monthly average temperature value for each province . In some cases we are not able to get valid values from some less detailed city council webs.

Considering that the temperature of each location varies in a similar way, although their values will be, in most of cases, different, the problem we face now is how to integrate those data sources in order to obtain the provinces average hourly temperature. Should we discard the latter data sources? Should we consider only the data sources with known values exactly each hour?

2.2 General architecture for data integration

There are two well known architectures for integrating data: Data Warehouses (DW) and Federated Database Systems (FDBS). Inmon defined a DW as “a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management’s decision-making process” [Inmon 02]. A DW is a database that stores a copy of the operational data with an optimized structure for query and analysis. A FDBS is formed by different component database systems; it provides integrated access to them: they co-operate with each other to produce consolidated answers to the queries defined over the FDBS. The FDBS has no data of its own unlike the DW. Queries are answered in the FDBS by accessing the component database systems. We have extended the Sheth & Larson five-level FDBS architecture [Sheth, Larson 90], which is very general and encompasses most of the previously existing architectures.

In order to carry out the integration process, it will be necessary to transfer from the data of the data sources, probably specified in different data models, to a common data model, that will be the used as the model to design the scheme of the warehouse. OWL is the ontology definition language we have chosen as the Common Data Model (CDM). It has been extended to support the adequate representation of spatial information (geographical, topological…) and the correct treatment of temporal information about all the data repositories in the system. We call this extension STOWL (Spatio-Temporal OWL).

Taking paper [Sheth, Larson 90] as point of departure, we propose the reference architecture in [Araque et al. 07]. The following relevant components of the architecture are outlinedbelow.

- Each data source will have a Native Schema expressed in its own data model, the data inherent to the source and the metadata (availability, temporal and spatial level detail…).
- In the Preintegration phase, the semantic enrichment of the data source‘s native schemas is made by the Conversion processor. In addition, the data source temporal and spatial metadata are used to enrich the data source schema with temporal and spatial properties. We obtain the component schema (CS) expressed in the CMD [Salguero et al. 08].
- From the CS, expressed in STOWL, the Negotiation processor generates the export schemas (ES) also expressed in STOWL. The ES represents the part of a component schema which is available for the Integrated Repository (IR) designer. For security or privacy reasons part of the CS can be hidden.
- The IR schema corresponds to the integration of multiple ES according to the IR designer needs. It is expressed in the CDM. This process is carried out by the Schema Integration Processor which suggests how to integrate the Export Schemas, thus helping to solve semantic heterogeneities, and defining the Extracting, Transforming and Loading processes.
- After the schema integration and once the IR schema is obtained, its maintenance and update will be necessary. This function is carried out by the Data Integration Processor. The algorithm explained in section 3 falls into this part of the architecture.

2.3 Temporal concepts

Due to the nature of the IR the annotation properties will usually refer to the temporal characteristics of data, so a set of annotation properties is defined to describe the sources according to some of the temporal concepts studied in [Araque et al. 06b].

STOWL defines, for instance, the Extraction Time of a change in a data source. The Extraction Time parameter can be defined as the time expended in extracting a data change from the source. Some examples of the temporal parameters which we consider of interest for the integration process are: Availability Window, Extraction Time, Transaction time, Storage Time, Temporal Granularity...

Definition 1. An historical value *h* is defined as $h = \langle v, d \rangle$ where *v* is the value given by the data source for the instant *d*. *H* denotes the set of historical observations.

Definition 2. The function V is a mapping $V : H \rightarrow \Re$ defined as $V(\langle v, d \rangle) = v$. The function D is a mapping $D : H \rightarrow \Re$ defined as $D(\langle v, d \rangle) = d$.

Definition 3. Let S^p be a data source consisting of a collection of historical observations about the feature p with the form $S^p = \langle h_1, ..., h_i \rangle$, where h_i corresponds to the ith historical value given by the data source for the parameter p. The granularity G of the source S for the parameter p is defined as

$$G(S^p) = \max\{g \,\big|\, g \leq \min\{D(h_j - h_i)\} \; \forall h_i, h_j \in H \; i < j \; g \in T\} \qquad (1)$$

where T is the set of temporal units (millisecond, day, year...). In other words, the granularity of a source for a parameter is the smallest time unit which is equal to or greater than the smallest interval between two consecutives changes in that source for that parameter. As a consequence, there can be only one valid value for each source, for each parameter at the same temporal granule.

Example. Given the data sources commented upon in the introductory section, the Granada airport and Motril sources have granularity of an “hour”. None of them are capable of giving more than one temperature value within the same hour. The granularity concept stands for the portion of the time the value has significance. The former data source has relevant values for each of its granules. The latter data source only has one relevant value every twelve granules. The granularity of the Granada city council’s source is a “minute”.

Sometimes, the granularity of a data source is also referred as its “level of temporal detail”. The smaller the granule the higher the temporal level of the detail of the source and the lower its granularity.

To make the algorithm more efficient we relax the constraints of the previous definition and we define the subgranularity concept G^* as

$$G^*(S^p) = \left\langle \begin{matrix} \min\{n \mid \exists n \cdot g \leq \min\{D(h_j - h_i)\} \;\; \forall h_i, h_j \in S^p, \, i < j, \, g \in T, n \in \mathrm{N}\}, \\ \max\{g \mid \exists n \cdot g \leq \min\{D(h_j - h_i)\} \;\; \forall h_i, h_j \in S^p, \, i < j, \, g \in T, \, n \in \mathrm{N}\} \end{matrix} \right\rangle \qquad (2)$$

In other words, the subgranularity can be set as multiple units of a granule (two seconds, ten days…). Furthermore, the subgranularity of a source for a given parameter is the lowest multiple of a time unit which better fits the minimum interval between two consecutive changes (two days instead of forty eight hours).

Following the running example presented in the introductory section, the airport data source has a subgranularity of an hour, the Granada City Council’s Web data source has a subgranularity of thirty minutes and Motril City Council’s Web has a subgranularity of twelve hours. In this case the subgranularity coincides with the changing rate of the sources because we suppose there are no missing values.

2.4 The Ordered Weighted Averaging operator (OWA)

The concepts in the previous section help us to find the values which should be integrated together. Once these values are determined, a solution based on the OWA operator is used to aggregate them. We shall introduce it briefly in this section.

The OWA operator was proposed by Yager in [Yager 88] as a new aggregation technique. It basically consists of performing a weighted average aggregation where the weights are not associated with the position the values hold, but the position they hold in the set after applying a particular ordering function to them.

Definition 4. An OWA operator of dimension n is a mapping $f : \Re^n \rightarrow \Re$ that has an associated n vector $W=[w^1\ w^2\ ...\ w^n]^T$ such that $w_i \in [0,1]$ and $\sum w_i = 1$. Furthermore $f(a_1,...a_n) = \sum w_j b_j$ where b_j is the jth largest of the a_i.

Yager also introduced some measures associated with an OWA operator [Yager 88]. One of these measures, which we consider of interest for this work, is the *orness* measurement, defined as

$$\text{Orness(W)} = \frac{1}{n-1}\sum_{i=1}^{n}((n-i)\cdot w_i) \qquad (3)$$

This measurement illustrates how the weighting vector values are displaced near the top (*Orness*(W) > 0.5) or near the bottom (*Orness*(W) < 0.5).

One of the main problems when using the OWA operator is the election of the weighting vector. In our case, the weighting vector is dynamically built every time an aggregating operation has to be performed for obtaining the value of each integrated granule. For this task we make use of a Basic Unit-interval Monotonic (BUM) function, introduced in [Yager 96]. A BUM function is a mapping f:[0,1]→[0,1] defined such that $f(0)=0$, $f(1)=1$ and if $x > y$ then $f(x) \geq f(y)$. Each component of a weighting vector of dimension n is obtained as

$$w_j = f\left(\frac{j}{n}\right) - f\left(\frac{j-1}{n}\right) \qquad (4)$$

Depending on the election of a specific BUM function, the *orness* of the weighting vector will vary between 0.5 (with $f(x) = x$) and 1 (with $f(x) = \{1 \text{ if } x=1, 0 \text{ otherwise}\}$), i.e. we can determine how the vector values are displaced to the top. We will use this function for building the weighting vector dynamically according to the number of available data sources at each instant.

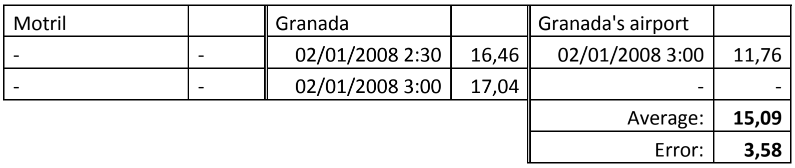
3 Data integration

In previous section, we argue that the usage of an OWA-based aggregation operator improves the overall quality of the resultant integrated information when dealing with data sources referring to measurements which evolve in a similar way. In this section the steps to perform this integration are detailed.

To measure the quality of the integration methods which are going to be described in this section we have defined three noisy sinusoidal functions which will represent the values of the data sources presented in the running example. In this way, the real integrated function can be calculated because all the values are known at every moment. We use this real data to compare the results obtained by each integration method. To simulate the temporal granularity of the data sources in the example, only one valid value is given for each granule of each data source in the example. We suppose there are no granules with missing values.

Given two or more data sources with a different temporal level of detail and the intended level of detail of the integrated data repository, there is basically two possibilities when performing the integration:

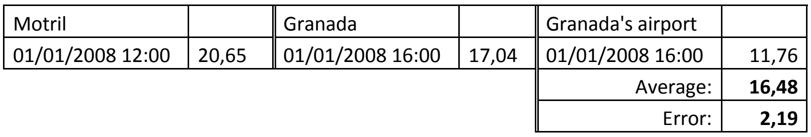
- All values averaging (AVA). Divide the time in granules with the same size as the granules in the integrated data repository and assign them the average value of all the values of the data sources within this period of time. This method, as illustrated in figure 1(a), has an significant problem: the result of the integration tends to the data source with the highest quantity of values. The resulting integrated value for the target granule [“2:30”-“3:30”] is the average value of the values in table 1 (rounded in figure 1(a)). The real value for “2:00” is 18.67, obtained as the average of the artificial sinusoidal functions described previously.



Motril		Granada		Granada's airport	
-	-	02/01/2008 2:30	16,46	02/01/2008 3:00	11,76
-	-	02/01/2008 3:00	17,04	-	-
				Average:	**15,09**
				Error:	**3,58**

Table 1: AVA-based aggregation for granule [02/01/2008 2:30 - 02/01/2008 3:30]

- Closest values averaging (CVA). Divide the time in granules with the same size as the granules in the integrated data repository and assign them the average value of the closest values of each data source to the centre of the granule, as illustrated in figure 1(b). The table 2 shows the values involved in the aggregation operation for the target granule [“2:30”-“3:30”].



Motril		Granada		Granada's airport	
01/01/2008 12:00	20,65	01/01/2008 16:00	17,04	01/01/2008 16:00	11,76
				Average:	**16,48**
				Error:	**2,19**

Table 2: CVA-based aggregation for granule [02/01/2008 2:30-02/01/2008 3:30]

Most of the usual IS packages use one of the previous approaches for aggregating the information. These techniques work well in most cases but there are some situations in which they fail. Given the granularity of the data sources containing the parameter which is going to be integrated and the desired granularity of this parameter’s values in the IR we can distinguish two different situations:

- The granularity of all the data sources containing the parameter to be integrated is lower than the desired granularity for the parameter in the IR. This means that all the data sources are capable of supplying at least one value for each of the integrated granules.

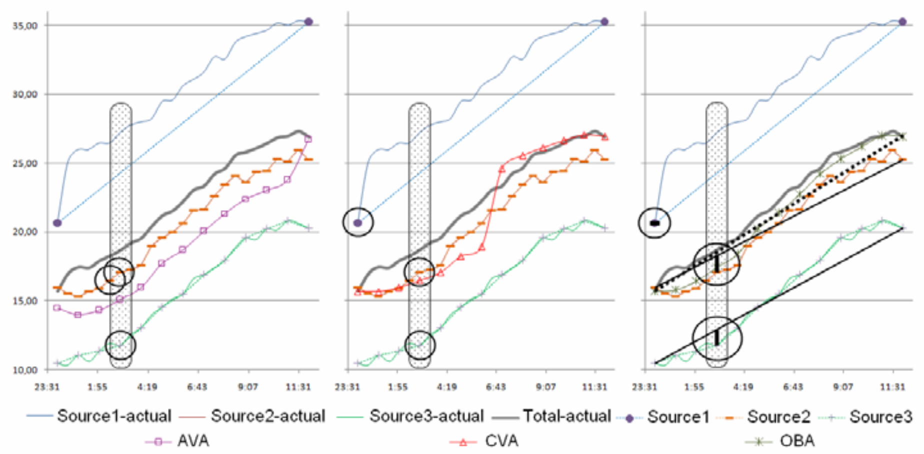


Figure 1: Comparison between the AVA, CVA and OWA-based aggregation techniques for the target granule ["2:30"-"3:30"].

- The granularity of some data sources is higher than the desired granularity for the parameter in the IR. This situation implies that there is at least one data source not capable of giving one value for each integrated granule.

In the former situation, both integration approaches give good results. It is supposed that there are enough values in all the data sources to discard those which introduce noise in the result. If using an AVA-based aggregation (ABA) operator it is possible to discard values from the data sources which contributes the larger quantity of values so as to prevent the result tending to those data sources. When using a CVA-based aggregation (CBA) operator the higher the quantity of values in the data sources the higher the possibility of finding a representative value for the integrated granule (closer to the integrated granule centre).

Although the CVA-based aggregation seems to be the most effective approach, in the latter situation it does not give very good results because the values of the data source with higher granularity contribute to many integrated granules. When the data sources to be integrated vary in a similar way the OWA-based aggregation (OBA) method proposed in this work provides a better solution.

Some data sources vary in a similar way when a variation in the values of one of them implies a similar variation in the values of the rest. This means that if a parameter in a data source is increased by ten units at some point, the value of the same parameter of the rest of the data sources have probably been increased in a similar way. This occurs in many situations and very often when designing DW based IS (integration of weather measurements of near locations, prices…).

Basically, the OBA operator acts over a set of time related historical values coming from all the data sources. After selecting the values to be integrated the OBA process is performed. Both steps are explained in the following section.

3.1 Step1: Selecting the values to be integrated

Each *Data Integration Processor* is responsible for undertaking the incremental capture of its corresponding data source and transforming the data to solve the semantic heterogeneities according to the integration rules obtained in the integration phase. In order to keep the integrated repository up to date without having to repeatedly query each data source, a set of algorithms for determining when they should be queried, according to their temporal characteristics, was proposed in [Araque et al. 06b].

The *Global Data Integrator Processor*, which takes the information from the data buffers provided by each *Data Integration Processor*, determines the sets of related data which should be integrated together (those in the same target granule) and passes it to the aggregation operator. The processor retrieves the temporal characteristics of the sources by accessing the STWOL data sources schemas [Salguero et al. 08] which have been extended to support this kind of metadata. All the changes detected in this data source are aggregated into one unique value according to the procedure introduced in the following section.

3.2 Step 2: The OWA-based Aggregation Operator

Once the values which influence the integrated granule are determined, an aggregation operation is needed in order to obtain a unique value corresponding to the integration of those values. In this case we propose the usage of an OWA-based operator. Below we explain the steps to build the required structures: the vector of values and the weighting vector.

It makes no sense to apply the OWA operator directly to the values to be integrated. In this case, the result would tend to the data source with the highest values. Instead, the importance (the order) of the values is going to be established by the amount of information the values provide about the tendency of the integrated function. In other words, the values are ordered according to the difference between the actual value and the value corresponding to the general shape of the function (low frequencies) at the same point.

The values to be integrated are obtained in the previous step. The weighting vector is calculated dynamically for each integrated granule according to a BUM-based function provided for each specific problem. The most complex task in this step is to obtain the order of the values. For this task the following algorithm is proposed. This algorithm is performed every time the calculation of an integrated value for an integrated granule is needed. It takes as the input the historical values related to the target integrated granule. In most cases, due to the fact that most of the data sources change at every granule unit, there will be at least one valid value for each data source corresponding to each integrated granule. On the other hand, as defined previously, the granularity of a data source does not imply that the data source must supply a valid value for each granule. In the event of existing missing values the algorithm still works reasonably well and better than the other techniques. The unique difference is that the BUM-based weighting vector would contain fewer items.

Firstly, the algorithm determines the historical values closest to the start, the end and the center of the integrated granule for each of the data sources. Then, for each historical value closest to the center of the granule a linear interpolated value is calculated according to the values corresponding to the start and the end of the granule (i.e. to the closest ones). The integrated value corresponds to the average of all the interpolated values plus the OWA-based aggregation of all the differences between the interpolated values and the actual known values. This way, the result is guided by the data sources which provide more information about the changes of the subjacent data at every point. In other words, the integrated function is the result of the arithmetic mean of the basic shape of the sources plus the difference between this and the actual value of the sources with the highest energy at every point.

Procedure name: OWA integration.

```
Input:
  H{s}{n}: nth historical value for data source s.
  Tstart: integrated granule start
  Tend: integrated granule end
  B: BUM function for building the weighting vector.
Output:
  I: integrated value.
Procedure:

  Granule_center = (Tend-Tstart)/2

  -- determine the closest value to the granule center for each source
  For i=0 to Count(H)
  Begin
    Closestsstart{i} = closest_value(H{i}, Tstart)
    Closestsend{i} = closest_value(H{i}, Tend)
    Closestcenter{i} = closest_value(H{i}, Granule_center)
  End

  -- for each data source
  For i=0 to Count(H)
  Begin
    interpolated{i} = linear_interpolation(D(Closestcenter{i}),
                      Closestsstart{i}, Closestsend{i})
    Interpolated_value = Interpolated_value + V(interpolated{i}) /
                         Count(H)

    Differences{i} = V(Closestcenter{i})-V(interpolated{i})
  End

  Weights = build_BUM_weighting_vector(B, Count(H))

  I = Interpolated_value + OWA(Differences, Weights)
```

According to the method previously described, firstly it is necessary to determine the previous and the next reference points according to the granularity of the data sources. In this case, the reference starting and ending points for the target granule are, respectively, “02/01/2008 00:00” and “02/01/2008 12:00”. These correspond to the previous and the next points of the data source with higher granularity with respect to the target granule center. For each data source the closest values to those points are retrieved (see table 3).

	Closest value to 02/01/2008 00:00		Closest value to 02/01/2008 12:00	
Motril	02/01/2008 00:00	20,65	02/01/2008 12:00	35,25
Granada	02/01/2008 00:00	15,96	02/01/2008 12:00	25,26
Granada's airport	02/01/2008 00:00	10,47	02/01/2008 12:00	20,25
Mean	02/01/2008 00:00	15,69	02/01/2008 12:00	26,92

Table 3: Closest values to higher granularity source limits

According to the mean values shown in table 3 a linear interpolation function between those mean values is defined as:

$$I(d,p_0,p_1)=V(p_0)+\frac{D(d)-D(p_0))(V(p_1)-V(p_0)}{D(p_1)-D(p_0)} \qquad (5)$$

Where V and D corresponds to the functions defined in section 2.2, p_0 and p_1 correspond respectively to the previous and next reference mean points of the corresponding data source. Note that these reference points belong to H, i.e. they have an associated value for the instant they represent. Considering the mean of all the starting and ending values, i.e. <02/01/2008 00:00, 15.69> and <02/01/2008 12:00, 26.92>, we would obtain a temperature of 18.5 for the "02/01/2008 03:00".

Once we have the base value for that point the OWA operator is used to improve it. The idea consists basically of getting the differences between this base value and the real values each data source provided for that point and performing an OWA aggregation where the weighting vector is built according to these differences. The problem is that in most cases the sources do not provide data for that specific point, so we use the closest one of each data source (column "A" in table 4). We suppose that the greater the difference between the interpolated value at those points and the real values given by the sources (absolute value of column "D" in table 4) the more relevant the information supplied by a data source at that point, so the higher its importance (absolute value of column "D" in table 4).

	A = Closest to "02/01/2008 03:00"	B = Interpolate(A)	D=A-B	Order ABS(D)	W=Associated Weight	D*W
Motril	02/01/2008 00:00, 20'65	20,65	0	3	$(1/3)^2$=0,11	0
Granada	02/01/2008 03:00, 17'04	18,29	-1,25	1	1-$(2/3)^2$=0,56	-0.69
Granada's airport	02/01/2008 03:00, 11'76	12,92	-1,16	2	$(2/3)^2$-$(1/3)^2$=0,33	-0.38
					Sum:	-1,08
					Mean interpolated:	18,5
					Total:	17,42
					Error:	**1,25**

Table 4: OWA-based aggregation for granule [02/01/2008 02:30-02/01/2008 03:30]

The selection of the underlying BUM functions depends on the specific problem. In this case, for sake of simplicity, we have chosen $f(x)=x^2$ as the underlying BUM function. More information about the BUM function is given in the following section.

It can easily be seen that the OWA-based aggregation obtains better results than the other two classical methods for the same point. In the following section the integration results of each method considering the entire data sources are discussed.

4 Experimental results

In order to determine the advantages of using the integration method presented in this work a case study is presented in this section.

Knowing the values of all the sources at every point allows us to determine the real integrated function and use it to compare the aggregation techniques presented in this work. It is not possible to use real data sources because we are unlikely to know the actual values between the values supplied by the data source and we cannot measure how well the aggregation techniques work.

Knowing the values of all the sources at every instant allows us to determine the real integrated function and use it to compare with the aggregation techniques presented in this work. It is not possible to use real data sources because we would not know the actual values between the values supplied by the data source and we cannot measure how well the aggregation techniques work.

In figure 2, a comparison between the CBA, ABA and OBA aggregation techniques is illustrated. The first three data series correspond to the data of the three data sources presented in the running example: the first of the series corresponds to the temperature in Motril whereas the second and third correspond to the temperature in Granada and Granada airport, respectively. Some noise has been introduced into the data in order to produce more realistic temperature measurements.

As explained before, to simulate the temporal characteristics of the data only some of the real data will be introduced into the integration algorithms. We suppose there are no missing values. The next three series in figure 2 correspond to this data.

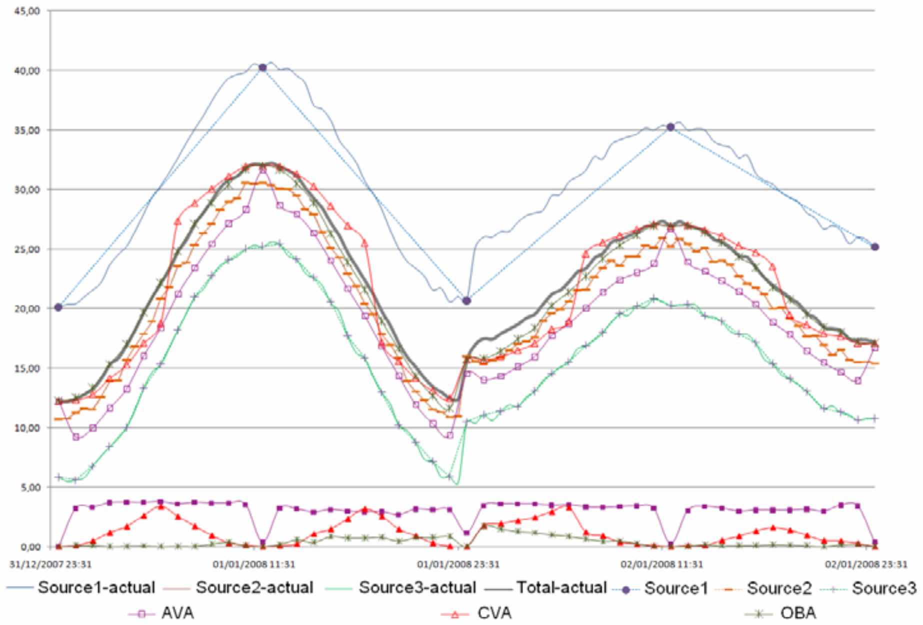


Figure 2: Results of the CVA, AVA and OWA-based aggregation techniques.

Depending on the granularity of the data source its series is more or less similar to the real one (marks have been drawn in the figure to indicate when the data is really available). The data source corresponding to Motril City Council's web is, for instance, far from realistic data because only one value is provided by this data source every twelve hours. On the other hand, the data source corresponding to Granada airport is very similar to the actual data because it provides data more frequently.

Once we have the data corresponding to the data sources the aggregation algorithms are carried out. Firstly, the series corresponding to the theoretical integration of the data sources, i.e. the integration performed knowing all the values at every point, is calculated. The next three data series correspond to the AVA-based aggregation, the CVA-based aggregation and the OWA-based aggregation.

As can be seen in the figure 2, the AVA-based aggregation method obtains the worst results. As expected, the integrated function using this technique tends to the data source which contributes more values, i.e. the source with the lowest granularity.

Both the CBA and the OBA obtain results which tend, as desired, to the theoretical integrated function. To compare the performance of each method three series are added to the chart consisting of the difference between the result of each integration method and the theoretical integrated function. These are shown in the bottom of figure 2. Both the CVA and the OWA-based method work well when there are many values available from the sources. This is the case, for instance, at 00:00 and at 12:00 of every day. All the data sources provide accurate values for those points.

The problem arises when the closest values supplied by the data sources are far from the integrated granule. At 06:00 or 18:00 the closest value of the data source corresponding to Motril City Council's web is six hours apart. This is the reason the

CVA-based aggregation method fails when dealing with these kind of data sources. On the other hand, this method obtains very good results when all the data sources provide data closest to the reference point. To take advantage of this situation the underlying BUM function would need to be modified according to this so as to improve the OBA method even more: use an underlying BUM function with a medium *orness* value when many of the sources provide actual data for the reference point (perform an arithmetic mean) and use an underlying BUM function with a higher *orness* value in the opposite situation in order to give importance to the sources which give more information.

5 Conclusions

In this paper, our work for the data integration of autonomous data sources, taking into account its temporal metadata properties, has been presented. For this integration process we have proposed an algorithm in order to obtain more precise data in the integrated repository. Improving the quality of the underlying data the quality of the decision-making processes based on this data is also enhanced.

In fact, the method proposed is focused on the situations where some of the data sources being integrated have a lower level of temporal detail than the desired detail for the integrated repository and evolve in a similar way. In these situations the improvement of the method with respect to the classical ones has been proven.

Acknowledgements

This work has been supported by the Andalucía Research Program under project GR2007/07-2 and by the Spanish Research Program under projects EA-2007-0228 and TIN2005-09098-C05-03.

References

[Abdulghafour et al. 94] Abdulghafour, M., Fellah, A. and Abidi, M. A.: "Fuzzy Logic-Based Data Integration: Theory and Applications"; Proc. of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, USA (1994), 151-160

[Araque et al. 06] Araque, F, Salguero, A. and Abad-Grau, MM.: "Application of data warehouse and decision support system in Soaring site recommendation"; Proc. of the 13th Conference on Information and Communication Technologies in Tourism (ENTER), Lausanne, Switzerland (2006), 308-319.

[Araque et al. 06b] Araque, F., Salguero, A., Delgado, C. and Samos, J.: "Algorithms for integrating temporal properties of data in Data Warehousing"; Proc. of the 8th Int. Conference on Enterprise Information Systems (ICEIS), Paphos, Cyprus (2006), 193-199

[Araque et al. 07] Araque, F., Salguero, A. and Delgado, C.: "Information System Architecture for Data Warehousing"; Proc. of the European Computing Conference (WSEAS), Springer Verlag, Athens, Greece (2007), 457-464

[Inmon 02] Inmon, W.H.: "Building the Data Warehouse"; John Wiley (2002)

[Russo 94] Russo, F. and Ramponi, G.: “Fuzzy methods for multisensor data fusion”; IEEE Transactions on instrumentation and measurement, 43, 2 (1994), 288-294

[Salguero et al. 08] Salguero, A., Araque, F. and Delgado, C.: “Using ontology metadata for data warehousing”; 10th Int. Conf. on Enterprise Information Systems (ICEIS), Barcelona, Spain (2008), 497-500

[Sheth, Larson 90] Sheth, A. and Larson, J.: “Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases”; ACM Computing Surveys, 22, 3 (1990), 183-236

[Xu 08] Xu, Z.: “On multi-period multi-attribute decision making”; Knowlege Based Systems, 21, 2 (2008), 164-171

[Xu, Yager 08] Xu, Z. and Yager, R.R.: “Dynamic intuitionistic fuzzy multi-attribute decision making”; International Journal of Approximate Reasoning, 48, 1 (2008), 246-262

[Yager 88] Yager, R.R.: “On ordered weighted averaging aggregation operators in multicriteria decision making”; IEEE Transactions on Systems, Man and Cybernetics, 18, 1 (1988), 183-190

[Yager 96] Yager, R.R.: “Quantifier guided aggregation using OWA operators”; International Journal of Intelligent Systems, 11, 1 (1996), 49-73