Visualizing the Availability of Temporally Structured Sensor Data

Authors: Daniel Nüst[1], Felix Bache[1], Arne Bröring[1],
Christoph Stasch[1], Simon Jirka[2]

1) Institute for Geoinformatics, University of Münster, Germany
2) 52° North Initiative for Geospatial Open Source Software

Abstract: With the development of new sensor technologies and the standardization of sensor data formats, sensor data from different sources becomes available for many applications. A crucial task is to get an overview on the spatial and temporal extent for which data is available before integrating the data into an application. This work presents an approach for accessing the necessary information about the availability of temporally structured sensor data from sensor web services. We show different kinds of data availability visualization. Based on the required values we specify a new generic sensor web service interface operation that constitutes the foundation for realizing the presented visualization methods.

INTRODUCTION

In recent years, sensor technology has been continuously improved. Thus, its usage in various domains is rapidly growing. Sensor applications range from human health services, public and home security, environmental monitoring, and precision agriculture to early warning systems (e.g. Shepherd, 2005). Within those applications the data gathered is often made available via standardized web services (e.g. Stasch, 2008 and Jirka, 2009).

An expressive visualization of such sensor data is a key tool for users to explore relations or anomalies within the data. Therefore, it is important to provide user-friendly and intuitive applications for sensor data visualization. However, before looking at the actual data, the user has to determine for which time instants or time periods sensor data is available. This metadata is important for any spatio-temporal application. A web service may serve real time but also historic data of different periods, even varying in temporal resolution. This variety may lead to bad user experience, misinterpretation, or failure in trying to access data. Thus, a client application has to support the visualization of data availability. This of course requires that also the sensor web services must provide a suitable interface for enabling such data availability visualizations.

The following use case underlines the need for such functionality. A climate researcher wants to calculate the change in the global temperature over the last 50 years. He discovers several web services that offer world-wide temperature data. To find out which of those services meet her requirements, she needs to know for which time periods, at which sampling rate, and for which phenomena data is available.

A basic textual approach showing all available measurements in a tabular fashion is unsuitable. First, the huge amounts of data gathered in modern sensor networks cannot be handled by humans in such way. It is cumbersome to compare many time stamps in typical date and time formats such as “2009-12-24T12:42:17+01:00”.

Second, the transmission of these timestamps requires huge bandwidth and processing capabilities because each measurement is processed by the server and transferred to the client. This limitation prevents powerful visualization alternatives, for example a simple timeline with all events (see figure 1), where data visually merges into blocks.

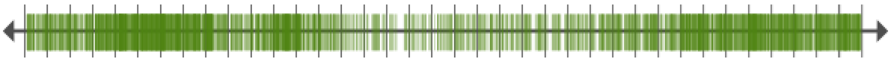


Figure 1: Simple timeline with every measurement event drawn as a small vertical dash. Huge processing capabilities are required.

Therefore, two kinds of aggregations are required: a graphical representation of accumulated data that increases comprehensibility (a gain both in speed and understanding) and a sensible data composition that reduces data volume. Of course, these abstractions lead to a loss in detail but this is a necessary trade-off in order to ensure a user friendly performance of the application.

We hypothesize that the user experience with sensor data can be enhanced with adapted visualizations. Furthermore, data accessibility increases with a special operation for sensor data services. In the remainder of this paper, we summarize related work and afterwards present visualization methods for sensor data availability. Based on these methods, we specify a sensor service operation. We conclude with a short evaluation and an outlook on future work.

RELATED WORK

Related fields of research comprise geovisualization, information visualization, and time-series analysis. Ahonen-Rainio, 2005 discusses the usage of multi-variate geographic metadata. TimeSearcher[1] and VizTree (see Lin, 2004) are popular examples sharing the importance of interactivity with the approach presented here. Work by Tominski, 2004 and Weber, 2001 show new and powerful approaches to time-series presentation. The diversity of methods for visualizing temporally structured data is presented by Aigner, 2008. Van Wijk, 1999 successfully uses a calendar-based visualization with their cluster analysis.

However, all of these methods target advanced scenarios with multiple variables (i.e. the actual values of measurements) or include pattern and trend analysis of spatio-temporal data. While these visualizations are often transferable to data availability visualization, they require or offer too much information for our particular task.

VISUALIZING SENSOR DATA AVAILABILITY

Visualization is a key to interpret and understand the information contained in the data (e.g. DiBiase, 1990). The data availability visualization is naturally performed as a preparatory search task to explore previously unknown data. It assists (optimally with high interaction) a user in choosing appropriate parameters for later data requests. Laymen and professionals are potential target audiences. We keep the design and interaction as simple as possible and visually appealing to both of them.

The task can be abstracted in the following way: Show a binary attribute, namely “data is available” or “no data”, as data points or intervals in an ordered one dimensional space (i.e. time). Optionally combine the attribute with geographic information. So, whether data IS present for known parameters, HOW MUCH data is retrievable, and HOW OFTEN, WHEN and WHERE data is available shall be possible to answer.

[1] http://www.cs.umd.edu/hcil/timesearcher/

The visualization methods require segmentation, i.e. an aggregation of data points into coherent temporal intervals. The spatial allocation of sensors and observed features is essential for spatial data analysis and can be added to all of the presented methods. In the next section, we outline and discuss methods for visualizing sensor data availability.

Visualization Methods

Timeline: A timeline is commonly used to manage a data display. There are various types with graphical variations such as linear or logarithmic scale, 2D or 3D to select the temporal interval of interest. Advanced timelines are based on aggregating the data into periods. Our design is a stacked time bar chart in which data availability of different features is displayed in blocks (see figure 2). The user can see for which intervals data entries are accessible. The density of values is shown with a variation in brightness. A lower brightness shows higher data density and vice versa. A steady lightness represents uniformly distributed measurements. On the left, the sensor data request parameters are displayed in a tree structure. The web service instance is at the highest level in the tree structure and the sensors are at the lowest level. In between, the observed feature and phenomenon are shown. On the right a set of horizontal bars show the periods of time for which data is provided with the newest data to the right outermost side. Edsall, 2005 points out to take the different cultural conceptualization of ordering into account. Our design is applicable for any horizontal or vertical ordering.

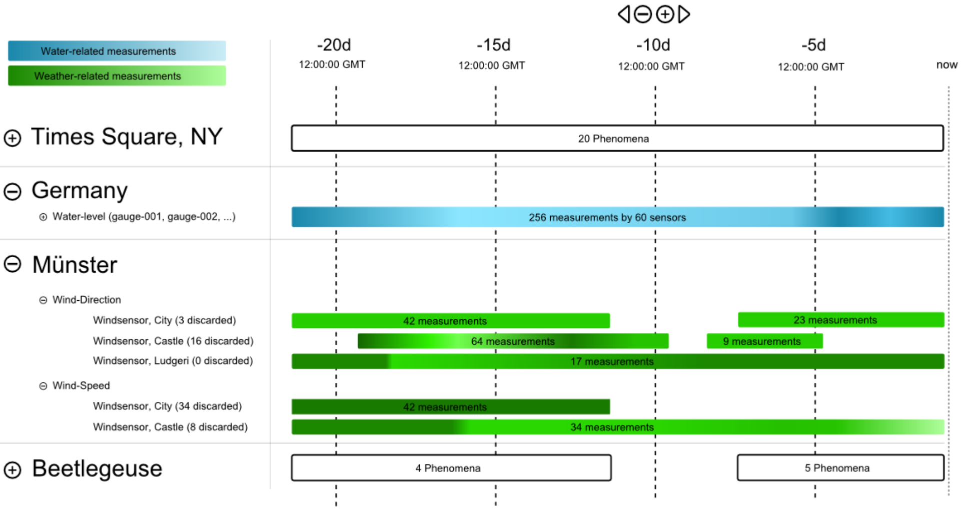


Figure 2: Stacked timelines with bars to the right and tree structure for a hierarchy of observed feature, phenomenon and sensors to the left. Uncolored bars combine collapsed branches.

Circular Bar Graph: The circular bar graph shows time intervals as bars on circular axes that arc around a central point in clockwise ordering. The axis of zero reference is oriented to the top (12 o'clock). Tick marks aligned with sequential labels show the units. The visual variable to identify start and end point of intervals is orientation (MacEachren, 1995). A user is able to differentiate periods because of the well known clock-style ordering. Additional information, in this case the count of data values for the respective period, facilitates comparing graphs. Any time period can be mapped to the 360 degrees of the polar axis.

A special case is certainly a twelve hour period where the graph resembles a common clock. This can be advantageous, as it is easy to understand. But it may be misleading if the period is arbitrary.

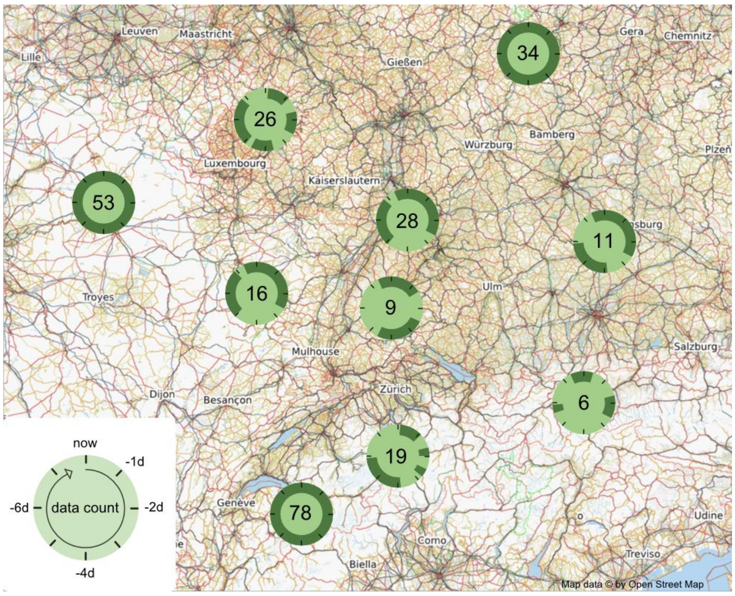
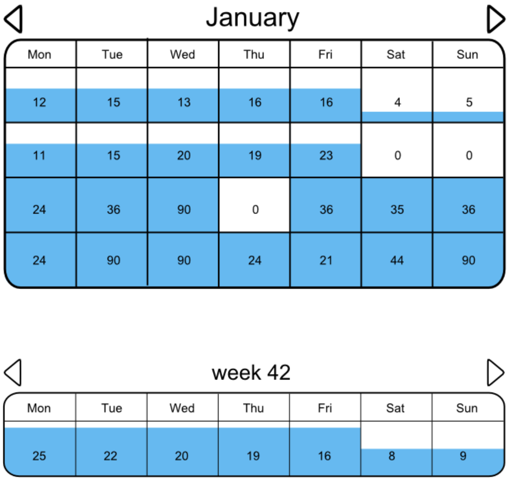


Figure 3: Circular bar graphs on a map with the data count inside the graph.

Figure 3 shows a combination of the circular bar graph with a map by using the origins as a representation for the location of an observed feature. The graph could be extended by scaling it proportionally to the amount of data, by using the polar distance to encode an additional parameter, or by stacking several rings of bars to visualize more than one phenomenon.

Calendar Sheet: This technique shows bar graphs per day placed on a calendar sheet. It takes advantage of a user's familiarity with the ordering of days on a calendar (as illustrated in figure 4).



January

Mon	Tue	Wed	Thu	Fri	Sat	Sun
12	15	13	16	16	4	5
11	15	20	19	23	0	0
24	36	90	0	36	35	36
24	90	90	24	21	44	90

week 42

Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	22	20	19	16	8	9

Figure 4: Calendar visualization with bars representing the data density for the day and the available count of measurements.

Within each day the height of the colored rectangle represents the proportion of data coverage and the count of measurements as numbers. In our example one measurement per hour is enough to “completely cover” a day. This visualization can be integrated into common calendar pop-ups for selecting begin date and end date of a period of interest. A disadvantage is the discrete and fixed categorization into the unit calendar day which forces splitting of periods spanning several days. Supplementary to the bar graphs, a timeline within the calendar may give the user not only a feeling for the amount of data but for the time spans for which data is available. The calendar metaphor can be expanded to a one day view where a timeline presentation is more expressive. It may also be extended by adding several phenomena in one calendar sheet for comparison.

The presented visualizations are based on a certain set of parameters which are encapsulated in the operation presented in the following section.

Interface Concept

Clients that produce visualizations and data providers communicate through standardized, interoperable web service interfaces. In the following, an operation consisting of request and response is defined which enables querying of sensor data availability. The design is kept simple and generic. The interface uses only identifiers and provides no additional metadata. It is abstract enough so it can be implemented either as a Remote Procedure Call using the WS-* technology stack (see Alonso, 2004) or as a RESTful interface (Fielding, 2000).

A segmentation algorithm on the server side requires input parameters. For starters we decided to include only one variable for a simple algorithm, the granularity. It is the maximum duration between measurements to be grouped together in an interval. Mielikäinen, 2006 presents advanced algorithms for automatic segmentation. Lone standing values could be discarded or treated specially. They are deliberately omitted at this point. If none of the optional parameters are set then all data is searched.

Figure 5 shows an UML notation of the request for data availability. The request contains two required parameters. ComputeHistogramData controls the calculation of the histogram and granularity is required by the segmentation algorithm as a duration element. Further, a client can request combinations of the parameters observed features, phenomena and sensors. A time period can be defined for the time span of interest.

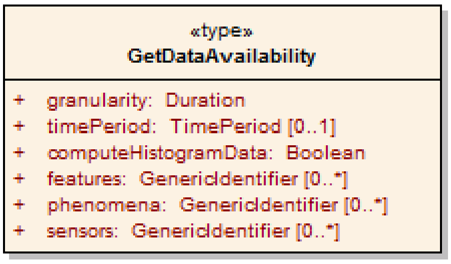

«type»
GetDataAvailability
+ granularity: Duration
+ timePeriod: TimePeriod [0..1]
+ computeHistogramData: Boolean
+ features: GenericIdentifier [0..*]
+ phenomena: GenericIdentifier [0..*]
+ sensors: GenericIdentifier [0..*]

Figure 5: UML diagram of GetDataAvailability operation.

Figure 6 shows a UML notation of the response for data availability. The properties features, phenomena and sensors have a many to many relationship. It changes depending on the view onto the system and leads to different hierarchical orderings. To keep the model simple, we group the parameters as attributes on the same level into a DataGroup, which is the basic entity of the response. A hierarchical ordering can be introduced at client runtime.

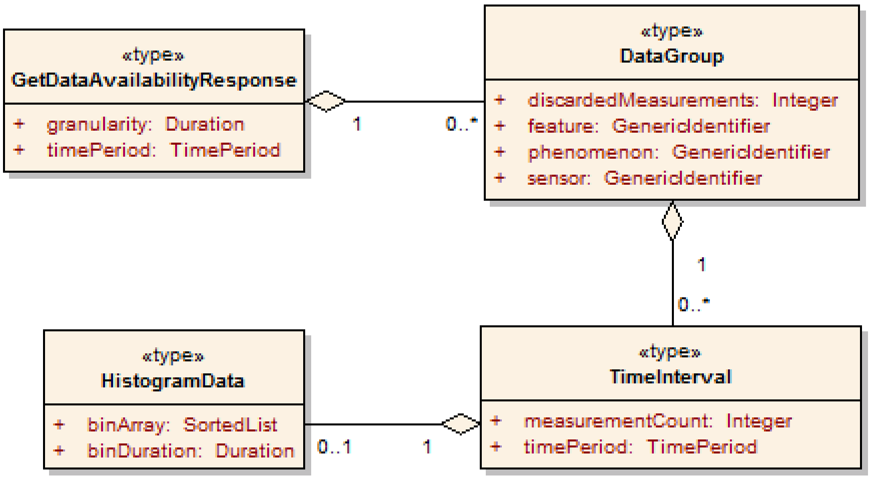


Figure 6: UML diagram of the GetDataAvailability operation response.

A DataGroup encapsulates the count of discarded measurements (to make the segmentation outcome transparent for a client) and a list of TimeIntervals. These can be seen as “fully covered” given the request’s granularity condition. TimeIntervals comprise the respective count of measurements, a time period with start data and end date. They can have a HistogramData element which provides information about the density and distribution of measurements within a TimeInterval and enable visualizations with gradients of color. It contains a subdivision of the interval into bins, which have a fixed temporal extend (binDuration) and the count of measurements within that sub-interval.

An inconclusive search does not have any DataGroups, whereas an existing combination of sensor, feature, and phenomenon is always present in a DataGroup even if there is no data in the specified time period. The required request parameters (granularity and timePeriod), are attributes of the response to facilitate asynchronous communication.

Special cases that are covered by this request and response are: querying of the available measurement count stretching across parameters or separately for all procedures; pure data availability querying with a Boolean result based on measurement count (true if bigger than zero, false otherwise).

CONCLUSION AND FURTHER WORK

The presented approach for improving the user interaction with sensor data services by supporting the selection of available sensor data consists of two steps. First, visualization methods for sensor data availability visualization are developed. Second, a generic sensor data service operation interface providing access to the required metadata is conceptualized. A special focus is put on the generality of the specified operations so that the underlying concepts can be easily transferred to different sensor service architecture approaches. Currently, the most commonly used sensor web architecture is the Sensor Web Enablement (SWE)[2] framework of the Open Geospatial Consortium (OGC). Within the SWE architecture the interoperable access to sensor data is covered by the Sensor Observation

[2] http://www.opengeospatial.org/projects/groups/sensorweb.

Service (SOS)[3]. Consequently the next step of our work will consist of extending the SOS interface by an operation supporting the data availability visualization. Within the current SOS interface, the existing operation GetFeatureOfInterestTime is limited to filter on a selected feature of interest and no mechanism to determine the amount of available data for specific spatial, temporal and thematic constellations is specified. We will suggest to the OGC to replace GetFeatureOfInterestTime with our data availability operation.

At the same time, we will work on the implementation of the presented visualization methods based on an existing open source client, the 52° North Thin SWE Client[4] . In order to enable this client extension we will enhance the 52° North SOS implementation by our data availability operation. In addition, usability tests with laymen and professionals will be conducted in order to evaluate the different data availability visualization methods we have presented.

For the future we consider further additions to the presented visualization methods. This includes animations, three dimensional views, and real-time updates. Regarding the interface an enhancement to four dimensions with spatial queries and spatial envelopes combined with temporal periods is expected. These additions are useful, for example to tackle use-cases involving mobile sensors. We plan to consider these use cases after our user evaluations.

ACKNOWLEDGEMENTS

The presented work is supported by the EC-funded project GENESIS (FP7-223996), as well as the 52° North Sensor Web Community. We thank the members of the Sensor Observation Service standardization working group at the Open Geospatial Consortium for fruitful discussions.

REFERENCES

Ahonen-Rainio, P., Kraak, M.-J., Towards multivariable visualization of metadata of geographic information. In J. Dykes, A. M. MacEachren, and M.-J. Kraak, (eds.). Exploring Geovisualization. Elsevier, Amsterdam: 611-626, 2005.

Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C., Visual methods for analyzing time-oriented data. Transactions on Visualization and Computer Graphics, 14, pp 47–60, 2008.

Alonso, G., Casati, F., Kuno, H., Machiraju, V., Web Services: Concepts, Architectures, Applications. Springer Verlag, 2004.

DiBiase, D., Visualization in the earth sciences. Earth and Mineral Sciences, 59, pp 13–18, 1990.

Edsall, R. M., Sidney, L. R., Applications of a cognitively informed framework for the design of interactive spatio-temporal representations. Elsevier, Amsterdam : 577–589, 2005.

Fielding, R. T., Architectural styles and the design of network-based software architectures. Ph.D. thesis. University of California, Irvine, 2000.

Jirka, S., Bröring, A., Stasch, C., Applying OGC Sensor Web Enablement to Risk Monitoring and Disaster Management. In GSDI 11 World Conference, Rotterdam, Netherlands, 2009.

Lin, J., Keogh, E., Lonardi, S., Lankford, J. P., Nystrom, D. M., Viztree: a tool for visually mining and monitoring massive time series databases. In VLDB ’04: Proceedings of the Thirtieth international conference on Very large data bases, pp 1269–1272. VLDB Endowment, 2004.

[3] http://www.opengeospatial.org/standards/sos.
[4] http://52north.org/maven/project-sites/swe/clients/Thin_SWE_Client/index.html.

MacEachren, A. M., How maps work : representation, visualization, and design. Guilford Press, New York, 1995.

Mielikäinen, T., Terzi, E., Tsaparas, P., Aggregating time partitions. In KDD ’06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, 2006. ISBN 1-59593-339-5, pp 347–356.

Shepherd, D., Kumar, S., Microsensor Applications. In S. S. Iyangar, R. R. Brooks (eds.). Distributed Sensor Networks. Chapman & Hall, 2005.

Stasch, C., Walkowski, A. C., Jirka, S., A Geosensor Network Architecture for Disaster Management based on Open Standards. In M. Ehlers, K. Behncke, F. W. Gerstengabe, F. Hillen, L. Koppers, L. Stroink, J. Wächter (eds.). Digital Earth Summit on Geoinformatics 2008: Tools for Climate Change Research, pp 54–59, 2008.

Tominski, C., Abello, J., Schumann, H., Axes-based visualizations with radial layouts. In SAC ’04: Proceedings of the 2004 ACM symposium on Applied computing. ACM, New York, NY, USA, 2004. ISBN 1-58113-812-1, pp 1242–1247.

Van Wijk, J. J., Van Selow, E. R., Cluster and calendar based visualization of time series data. In INFOVIS ’99: Proceedings of the 1999 IEEE Symposium on Information Visualization. IEEE Computer Society, Washington, DC, USA, 1999. ISBN 0-7695-0431-0.

Weber, M., Alexa, M., Müller, W., Visualizing time-series on spirals. In INFOVIS ’01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS’01). IEEE Computer Society, Washington, DC, USA, 2001. ISBN 0-7695-1342-5.