Time Series in VOEvents

John M. Brewer
San Francisco State University, University of California, Berkeley
Joshua S. Bloom
University of California, Berkeley

Here we present the SimpleTimeseries semantics and an XML realization, intended to serve as the standard mechanism for representing time series data within VOEvent. In addition to marking up traditional light curves (flux or magnitude changes with time), SimpleTimeseries has been designed to allow publishing of phased light curves, space motion, Doppler shift, and even gravity wave strain measurements with minimal software investment. It is relatively simple to construct and even read and yet can be strictly validated by standard parsers.

Motivation

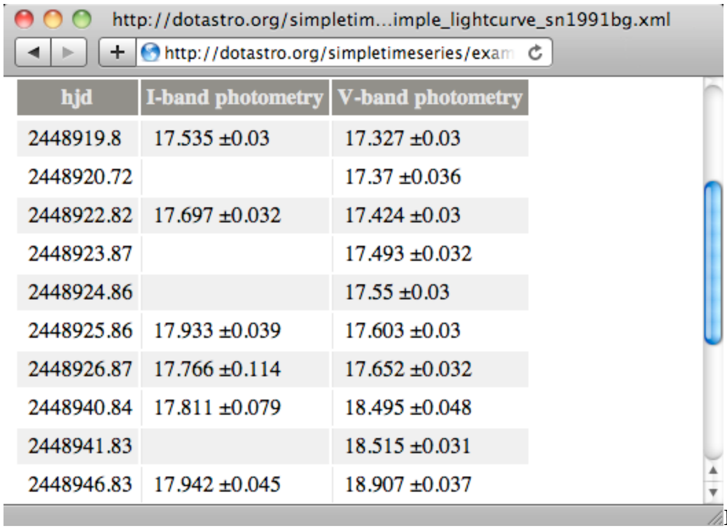
Synoptic surveys are transforming astronomical data collection from static photoshoots of the night's sky to full-scale cinematographic productions. The ensuing data deluge presents new opportunities for scientific exploration of transients and variable stars but also new challenges. A crucial component to the efficacy of these pursuits is the timely dissemination of new discoveries.

Thankfully, VOEvent[1] provides a standard messaging protocol to describe celestial transients at an instance of time. However, for the recipients of event discovery to use their precious telescope time to follow up sources of interest to them, increasingly they will place the onus on the discovery engines to publish light curves of the events. These light curves -- and the ones created from followup observations -- will serve as cornerstone of classification and inference.

The Berkeley Transient Classification Pipeline (TCP)[2,3], currently working with data from the Palomar Transients Factory (PTF)[4], is a project whose goal is rapid classification of astronomical transients from image difference data. Data captured at the telescope is sent through an image subtraction pipeline and light curves of any changing sources are forwarded on to TCP. After classifications are made, interesting candidates are sent on to a follow-up marshal that decides wether and how to allocate additional resources. The SimpleTimeseries XML format was designed, in part, with these needs in mind. Recognizing that a format that could handle more than just light curves would also benefit the larger astronomical community, we worked with the VOEvent working group to make our nascent format a standard.

Overarching Goal: Simplicity and Applicability

Time series have been published for centuries, but there is, as of yet, no canonical representation. Ten different astronomers are likely to publish nearly identical light curves in at least 10 slightly different formats. To manage and react to the enormous expected data flows, a predictable and standard time series format is crucial. Custom programming should not be necessary in order to parse the data stream from each new provider.



hjd	I-band photometry	V-band photometry
2448919.8	17.535 ±0.03	17.327 ±0.03
2448920.72		17.37 ±0.036
2448922.82	17.697 ±0.032	17.424 ±0.03
2448923.87		17.493 ±0.032
2448924.86		17.55 ±0.03
2448925.86	17.933 ±0.039	17.603 ±0.03
2448926.87	17.766 ±0.114	17.652 ±0.032
2448940.84	17.811 ±0.079	18.495 ±0.048
2448941.83		18.515 ±0.031
2448946.83	17.942 ±0.045	18.907 ±0.037

Figure 1. Sample output of a light curve in SimpleTimeseries format when viewed in a web browser. An XSL stylesheet could also render a plot of the light curve in addition to the data table.

Goals

Based on our experience with TCP and conversations with the VOEvent working group, we identified several key requirements for a time series format:

- Human and machine readable
- Light weight and easy to implement
- Extensible for time series data beyond light curves
- Integrates with existing standards
- Simple machine validation of data products

Efforts such as the IVOA Spectral Data Model [5] provide a way to represent time series. However, by allowing a high degree of flexibility in data representation, custom coding is required to ensure data integrity for each new data publisher. SimpleTimeseries provides a flexible model for time series, but in such a way as to enforce consistency and ease machine readability via the XML schema. By using XML it is also possible to provide style sheets to display the data in multiple human readable formats.

A SimpleTimeseries Light Curve

At the time of this writing, SimpleTimeseries is undergoing final revisions to better fit within VOEvent and comply with best practices established by the VO. The example below provides a good overview of how you can use SimpleTimeseries, but for the most up-to-date schema, please visit http://dotastro.org/simpletimeseries.

Structure

SimpleTimeseries is an XML format consisting of four major components:

- Time system (`TIMESYS`)
- Band Passes or Filters (`BAND`)
- Field Descriptions (`FIELD`)
- Time Series Data (`SERIES, ELEM, TIME`)

The most complex component is the time series data, though the only required axis in a `SERIES` is `TIME`. The first three items can be thought of as all of the meta-information which describes your data. This enables the data to be described by a relatively straight forward and space-conserving format. Each of the meta-data elements itself is described by Virtual Observatory Uniform Content Descriptors (UCD)[6]. In general, machine readable data are contained in attributes, and human readable data are stored as element values.

TIMESYS

The `TIMESYS` element is a container which holds the definition of the time system. This includes the type of time system used (ie barycentric Julian days), any zero-point offset, the units of time used in the data (ie days, seconds, etc.), default integration time, wether the data is period folded rather than using absolute times, the time scale, and the reference location for the time.

```
 <TIMESYS>
    <TimeType ucd="time;pos.frame;pos.heliocentric" unit="day">
       hjd
    </TimeType>
    <TimeZero ucd="time.epoch;arith.zp" unit="day">0</TimeZero>
    <TimeUnits ucd='time.epoch' datatype='float' unit='day' />
    <TimeWidthDefault ucd="time.period" unit="seconds">
       10.0
    </TimeWidthDefault>
    <TimeSystem ucd="frame.time.scale">TDB</TimeSystem>
</TIMESYS>
```

BAND

For most time series, the band pass in which the data was recorded is almost more important than the data itself. All `FIELD` elements require a reference to a `BAND`.

```
<BAND ucd="instr.filter;em.opt" bandid="I"
   description="The Johnson-Cousins I-band">I</BAND>
<BAND ucd="instr.filter;em.opt" bandid="V">V</BAND>
```

The required UCD provides standard machine readable information. The value of the element is a human readable label, and a brief description can be stored in the description attribute. An optional `uri` attribute can also be used to link to a more thorough treatment of the filter characteristics.

FIELD

Data in the series is identified by a reference to a `FIELD` not by its position in a table. This reference makes it trivial for a machine to quickly and accurately identify the data, its type, and the band pass it was recorded in.

```
<FIELD fld="imag" bandid="I" ucd="opt;phot;i" datatype="float"
   unit="mag">I-band photometry</FIELD>
<FIELD fld="vmag" bandid="V" ucd="opt;phot;v" datatype="float"
   unit="mag">V-band photometry</FIELD>
```

SERIES

Like `TIMESYS`, `SERIES` is just a container. Within `SERIES` lie one or more `ELEM`s which hold data relating to a particular timestamp. `ELEM` in turn contains the mandatory `TIME` element and zero or more values, each of which must reference a `FIELD`. A single `FIELD` may only be referenced once in each `ELEM`. Unlike a traditional table, the field references make it necessary to include only data values which were actually recorded. This saves space and reduces the proliferation of 'no data' proxies (such as 9999.99) which currently pepper astronomical time series.

```
<SERIES>
<ELEM row='1'>
   <TIME><T>2448919.8</T></TIME>
   <MAG fld="imag"><VAL>17.535</VAL><ERR>0.03</ERR></MAG>
```

```
    <MAG fld="vmag"><VAL>17.327</VAL><ERR>0.03</ERR></MAG>
</ELEM>
<ELEM row='2'>
    <TIME><T>2448922.82</T></TIME>
    <MAG fld="imag"><VAL>17.697</VAL><ERR>0.032</ERR></MAG>
    <MAG fld="vmag"><VAL>17.424</VAL><ERR>0.03</ERR></MAG>
</ELEM>
<ELEM row='2.5'>
    <TIME><T>2448923.87</T></TIME>
    <MAG fld="vmag"><VAL>17.493</VAL><ERR>0.032</ERR></MAG>
</ELEM>
<ELEM row='3'>
    <TIME><T>2448925.86</T></TIME>
    <MAG fld="imag"><VAL>17.933</VAL><ERR>0.039</ERR></MAG>
    <MAG fld="vmag"><VAL>17.603</VAL><ERR>0.03</ERR></MAG>
</ELEM>
...
```

Value elements can be one of position (`POS_EQ` and `POS_LB`), double precision floating point value (`MAG`, `FLUX`, and `DVAL`), or string (`NOTE`). The position elements for instance, group RA and DEC with their uncertainties to aid readability. Each of the numerical value elements can have 0, 1, or 2 error values and can be designated as upper or lower limits in addition to measured values.

```
<MAG fld="vmag" limit="upper">
    <VAL>17.535</VAL>
    <ERR>0.03</ERR><ERR>-0.06</ERR>
</MAG>
```

The `NOTE` element is designed to allow attaching human readable notes about data quality, authorship, seeing, etc. to the data. An integer code can also be included in the `NOTE` to aid machine readability.

Conclusion

The strength of SimpleTimeseries lies in its singular focus, simplicity in construction, and strictness in validation. As long as there is a time axis, it will be easy to publish and consume everything from transient light curves, the apparent motion of an asteroid, a gravity wave strain measurement of a neutron star coalescence event, or even just a list of upcoming observing times. The inclusion of SimpleTimeseries in the VOEvent standard enables a richer understanding of the event without the need for human interaction or extensive custom programming.

Acknowledgements

We would like to thank A. Allen, B. Denny, P. Kubánek, A. Rots, R. Seaman, D. Starr, R. Williams, and the participants of the HotWired II VOEvent Working group meeting for their invaluable assistance. Their contributions have made SimpleTimeseries into a robust and easy to use time series standard.

References

[1] Seaman et al. Sky Event Reporting Metadata (VOEvent) Version 1.1 http://ivoa.net/Documents/latest/VOEvent.html (http://voevent.org)

[2] Bloom et al. Towards a real-time transient classification engine. Astronomische Nachrichten (2008) vol. 329 pp. 284

[3] Starr et al. Real-time Transient Classification and Broadcast Pipeline. Astronomical Data Analysis Software and Systems ASP Conference Series (2008) vol. 394 pp. 609

[4] Law et al. The Palomar Transient Factory: System Overview, Performance and First Results. arXiv (2009) vol. astro-ph.IM

[5] McDowell et al. IVOA Spectral Data Model Version 1.03 http://www.ivoa.net/Documents/latest/SpectrumDM.html

[6] Martinez et al. The UCD1+ Controlled Vocabulary Version 1.23
http://www.ivoa.net/Documents/latest/UCDlist.html