OpyenXES: A Complete Python Library for the eXtensible Event Stream Standard

Hernan Valdivieso, Wai Lam Jonathan Lee,
Jorge Munoz-Gama, and Marcos Sepúlveda

Computer Science Department, School of Engineering
Pontificia Universidad Católica de Chile, Santiago, Chile
{hfvaldivieso,walee,jmun}@uc.cl
{marcos}@ing.puc.cl

Abstract. There has been a spectacular growth in the availability of event data. To transport, store and exchange these event data, the *eXtensible Event Stream* (XES) has become the acknowledged standardization due to its simplicity, flexibility, extensibility and expressivity. Currently, the *OpenXES* library exists as the popular open-source Java implementation of the XES standard. However, despite the gaining popularity of Python as the core programming language for data science, there has yet to exist a complete and open-source implementation of the standard in the language. This paper presents *OpyenXES* as a complete and open-source implementation of the XES standard in Python. This opens up the rich portfolio of Python packages for data science to researchers and practitioners in the field of business process management.

Keywords: event log, XES, Python, open-source, data science

1 Significance to the BPM Field

Nowadays we live in the *Big Data* era. Massive amounts of event data about our daily processes are recorded by information systems and electronic devices alike in the so-called *Internet of Events* [4]. For process stakeholders, one main goal is to turn event data into real value, e.g., improving the efficiency of the process. Under this context, a new format for event data has appeared: the *eXtensible Event Stream (XES)* [2]. XES is an XML-based standard for event logs and its purpose is to provide a generally-acknowledged format for the interchange of event log data between tools and application domains [2]. In 2016, the IEEE approved XES as the standard for achieving interoperability in event logs and event streams (1849-2016) [3]. Process-oriented data science has achieved immense success in the recent years, especially in the field of process mining. It has also led to the development of a rich set of tools and software, e.g., ProM [8] in the research field. One key component of these tools is having the full support for importing and exporting event data in the XES format. This has been achieved by *OpenXES*, the open source implementation of the XES standard in the Java programming language. Moreover, knowing the level of support for the

F. Casati et al. (Eds.): Proceedings of the Dissertation Award and Demonstration, Industrial Track at BPM 2018, CEUR-WS.org, 2018.

XES standard is such an important issue that one of the recent achievements of the XES working group is the approval of the XES certification proposal. With this, software can be certified for the level of support for the XES standard. At the same time, the popularity of Python as a programming language for data science has been rising in the recent years [6]. This has been due to the open source nature of the community, the collection of packages oriented towards data science, e.g., NumPy, SciPy, pandas, and matplotlib, and its high productivity for prototyping and building small and reusable systems. In fact, the code implementation of many recent research papers are in Python for precisely these reason. However, despite the large collection of useful tools for data science in the Python environment, this adoption trend is largely unseen in the field of business process management. We believe one of the key reasons is the lack of support in handling event data in the XES format. While there have been some previous attempts to fill this gap, e.g., simple Python scripts that support parts of the XES standard, or complex suites that are difficult to be used in standalone prototyping, there has yet to be a full implementation of the standard. In R, there are similar support for XES files with the BupaR packages [5]. However, it does not support the full XES standard.

In this work we make a serious step towards that direction by presenting *OpyenXES*, a complete open-source Python library for the XES standard. The remainder of the paper is structured as follows: Section 2 presents the library and the principles behind it. Section 3 illustrates the applicability of the library with a set of different examples. Section 4 presents the links to access the library, the source-code, and the documentation. Finally, Section 5 concludes the paper.

2 OpyenXES: The Library

OpyenXES is a Python library to support the creation, analysis, and storing of event data satisfying the XES standard. OpyenXES was created with the following principles in mind:

- ▷ *Python*: OpyenXES was implemented in Python for Python. It was designed to be programmed as any Python script, using typical Python syntax, e.g., iterate the traces of a log using 'for', or add a new event using 'append'. Moreover, it is registered under the Python Package Index (PyPI) so that it is pip installable.
- ▷ *Complete*: Unlike other Python scripts for XES, OpyenXES includes all the elements defined in the XES standard [3], such as 'classifiers' and 'extensions'. Figure 1 shows the overview of the main components of the library. Software using OpyenXES should have the same XES certification level as current software using OpenXES.
- ▷ *Open*: OpyenXES was created following the Open Science Principle, like other related tools such as OpenXES or ProM. Therefore, both the library and the source code are publicly available (cf. Section 4). This opens the door to the data science community to fork, correct, and improve the library

e.g., propose more efficient DB-based implementations for the storage and processing of the XES logs [7].

- ▷ *OpenXES Compatible*: The state-of-the-art implementation for XES is the OpenXES open-source Java library [1]. In order to ease the transition from one library to another (and the other way around), OpyenXES has preserved both the naming and the package structure of the more mature (cf. Figure 1). A researcher familiar with one library would be able to implement programs using the other library with little effort. Notice that, the name itself (OpyenXES) is an homage paid to and its creators and maintainers for their selfless work.
- ▷ *Combinable*: One of the main strength of Python is its data science and machine learning libraries (NumPy, SciPy, pandas, matplotlib, jupyter, scikit learn, ...). OpyenXES was designed to be combined with such libraries in order to realize its potential as a data science tool.

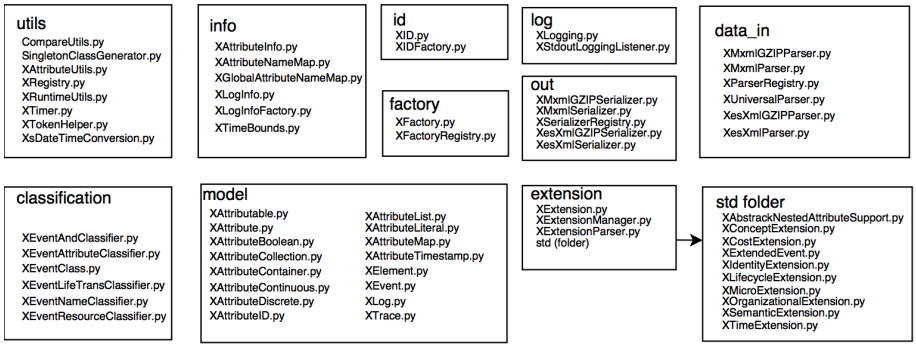

Fig. 1. Overview of the main components of OpyenXES.

3 Maturity and Use Cases

The following set of examples illustrate the maturity, usefulness, and applicability of OpyenXES for process-oriented data science and its combination with other popular Python libraries. Due to space limitations, the source codes of all the examples are available in `https://github.com/opyenxes/OpyenXes/tree/master/example`.

- ▷ *Log Anonymization*: An event log is iterated through to remove any information about the resources involved in the cases and the events.
- ▷ *CSV to XES*: Event data stored in a Comma Separated Value file (.csv) is converted into a XES event log file, including the date transformation into a XES compatible format.

- ▷ *Random Log*: A log is created from scratch using random values.
- ▷ *Filter Variants*: A new event log is generated by only preserving one of the traces from all the traces with the same sequences of activities, i.e., only the different variants of the original event log.
- ▷ *Alpha Algorithm*: The classic process mining discovery algorithm Alpha Algorithm [4] is easily implemented in Python using OpyenXES to generate the footprints.
- ▷ *Reporting Statistics*: An event log is iterated through to obtain statistics about the number of activities performed by each resource, and the results are graphically plotted using *matplotlib* library.
- ▷ *Trace Clustering*: OpyenXES is combined with the popular machine learning Python library *sci-kit learn* to clustering the traces using the *k-means* algorithm.

These examples only tease in an instructive way the full potential of OpyenXES and Python for process-oriented data science, opening the door to more complex operations such as: k-anonymity, tool-specific format transformation, Python specific libraries for CSV or JSON processing, event log simulation, domain-specific filters, other process mining algorithms, or the combination with well known data science python libraries.

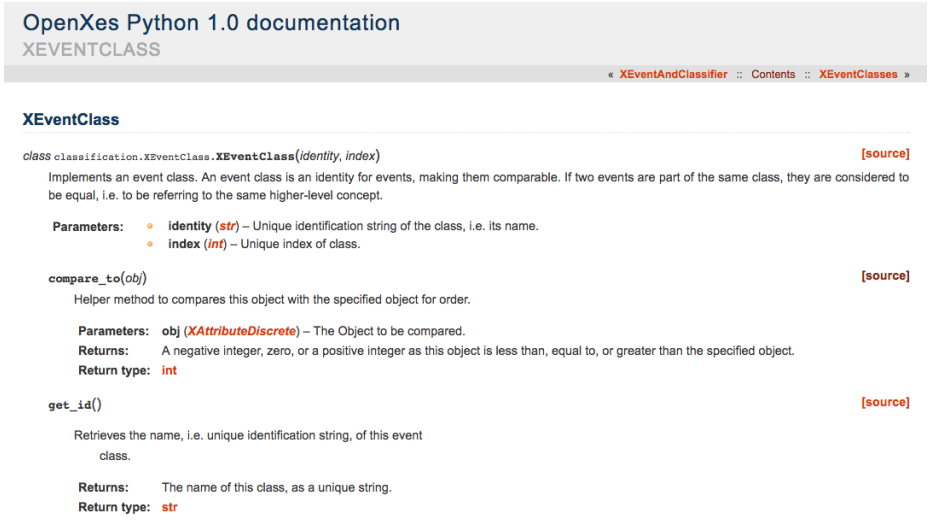

Fig. 2. Documentation and source code of OpyenXES.

4 Download, Screencast, and Links

The library is available as a GitHub public repository under the organization *opyenxes* (`https://github.com/opyenxes`) opening the door to be forked, ex-

tended, or improved by the community. The source code is also available in the repository. The library is completely documented in `http://opyenxes.readthedocs.io/en/latest/?badge=latest` (cf. Figure 2). A screencast illustrating the main features of *OpyenXES* is available in `www.processmininguc.com/tools`. Finally, examples showcasing the use of the library are available in `https://github.com/opyenxes/OpyenXes/tree/2018-bpm-demo/example`.

5 Conclusions

This paper presented OpyenXES, a Python library for the *eXtensible Event Stream* (XES), a format used for the interchange of event log data between tools and application domains. The library includes all the elements of the standard (such as classifiers or extensions), it is open-source and open to the community, and it respects the same naming and package architecture than the state-of-the-art Java library, easing the transition from one library to the other. We believe that OpyenXES could trigger the interest of a new set of data scientists for research in BPM and process mining.

Acknowledgments. This work is partially supported by *CONICYT-PCHA / Doctorado Nacional / 2017-21170612*, *Vicerrectoría de Investigación de la Pontificia Universidad Católica de Chile / Concurso Investigación Pregrado 2017*, and by the *Departamento de Ciencias de la Computación UC / Fond-DCC-2017-0001*. The authors would like to thank the members of the IEEE Task Force on Process Mining XES Working Group and the creators and mantainers of OpenXES Java Library for their selfless efforts.

References

1. OpenXes, `http://www.xes-standard.org/openxes/start`
2. XES Web, `http://www.xes-standard.org`
3. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. IEEE Std 1849-2016 pp. 1–50 (Nov 2016)
4. van der Aalst, W.M.P.: Process Mining - Data Science in Action. Springer (2016)
5. Janssenswillen, G., Depaire, B.: bupar: Business process analysis in R. In: Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with (BPM 2017). (2017)
6. Puget, J.F.: What Language is Best for Machine Learning and Data Science (2016), `https://www.ibm.com/developerworks/community/blogs/jfp/entry/What_Language_Is_Best_For_Machine_Learning_And_Data_Science`
7. Syamsiyah, A., van Dongen, B.F., van der Aalst, W.M.P.: DB-XES: enabling process discovery in the large. In: SIMPDA. CEUR Workshop Proceedings, vol. 1757, pp. 63–77. CEUR-WS.org (2016)
8. Verbeek, E., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Prom 6: The process mining toolkit. In: Proceedings of the Business Process Management 2010 Demonstration Track, Hoboken, NJ, USA, September 14-16, 2010 (2010)