ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV047

STATUS OF HIGH LEVEL APPLICATION DEVELOPMENT FOR HEPS*

Xiaohan Lu†, Qiang Ye, Hongfei Ji, Yi Jiao, Jingyi Li, Cai Meng, Yuemei Peng, Gang Xu, Yaliang Zhao
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China

Abstract

The High Energy Photon Source (HEPS) is a 6 GeV, 1.3 km, ultralow emittance ring-based light source in China. The construction started in 2019. In this year, the development of beam commissioning software of HEPS started. It was planned to use EPICS as the control system and Python as the main development tools for high level applications (HLAs). Python has very rich and mature modules to meet the challenging requirements of HEPS commissioning and operation, such as PyQt5 for graphical user interface (GUI) application development, PyEPICS and P4P for communicating with EPICS. A client-server framework was proposed for online calculations and always-running programs. Model based control is also one important design criteria, all the online commissioning software should be easily connected to a powerful virtual accelerator (VA) for comparison and predicting actual beam behaviour. It was planned to use elegant and Ocelot as the core calculation model of VA.

INTRODUCTION

The High Energy Photon Source (HEPS) is a 6 GeV, 1.3km, green-field 4th generation storage ring light source [1]. For this light source, the lattice design and related physics studies were basically finished [2-6], and the construction started in mid-2019. The machine commissioning of LINAC is expected to start in mid-2022. By this time, the high level applications (HLAs) for beam commissioning are indispensable.

To achieve this goal, the development of HLAs started this year. Before the start of development, we investigated the HLA schemes of almost all the running light sources. Most of them use the Matlab Middle Layer Toolkit (MML) [7, 8] as the main commissioning tools, and part of them developed new specific commissioning tools, such python-based aphla of NSLSII [9], and python-based HLA of Sirius [10]. Recently, python becomes more and more popular in every walk of life, due to its powerful modules and easy-to-learn characteristic. And more and more labs use python as the main development tools for control software and data analysis tool. In consideration of future development, economic and time cost, for the HEPS python was chosen as the main development tools, and EPICS as the control system. Python has very rich modules to meet the demands of HLA. PyQt5 was used to develop GUI app, will be update to pyqt6 in near future, PyEPICS and P4P were used as the communication tools. The HLA of HEPS is model-based, we decided to use Elegant [11] and Ocelot [12] as the core calculation models. A client-server framework was proposed for online calculations and always-running programs. All the online commissioning software should be easily connected to a powerful virtual accelerator (VA) for comparison and predicting actual beam behaviour.

APPLICATIONS FRAMEWORK

The development of HLA is collaborative work between the control group and physics group. The whole control system framework is shown in Fig. 1, all the applications were classified into three categories: parameter display and hardware controls, algorithm-based applications for tuning or measurements, model-based applications. The accelerator physics group is responsible for the development of the algorithm-based and model-based applications (the green part in Fig.1), and the control group is responsible for the development of the rest of applications and the construction of the low-level control applications (the blue part in Fig. 1).

The control group chose Control System Studio (CSS) [13] and python display management (PyDM) to build the parameter display and hardware control applications. The accelerator physics group is responsible to develop a brand new platform python accelerator physics application set (Pyapas) to build the HLA, as shown in Fig.2 with the aim of including all necessary tools and modules.

Pyapas is not only a set of applications, but also designed as a platform for the rapid development of HLA. Pyapas learns design philosophy from PyDM [14] and OpenXAL [15]. There are three principles of Pyapas, the first one is that all the applications are physical quantity based. The parameters we work with should be physical quantity. For example, when we do the orbit correction, we should change the magnetic field rather than the current of the correctors. The second one is that most of the applications should be server-based, the calculation process could be running in the background. The third one is that the applications should be model-based, all the applications could be connected to the VA, use the simulation results to tune and predict the behaviour of the beam.

According to the design principles, a clear roadmap was made to develop Pyapas. The drag-and-drop way should be implemented to develop GUI applications quickly. To develop server-based application, a robust and convenient client-server framework is indispensable. The model-based applications and VA require powerful physical models, for the moment we decide to use the exist mature simulation code Elegant and Ocelot as the core physical model. To

*Work supported by NSFC (12005239)
† luxh@ihep.ac.cn

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV047

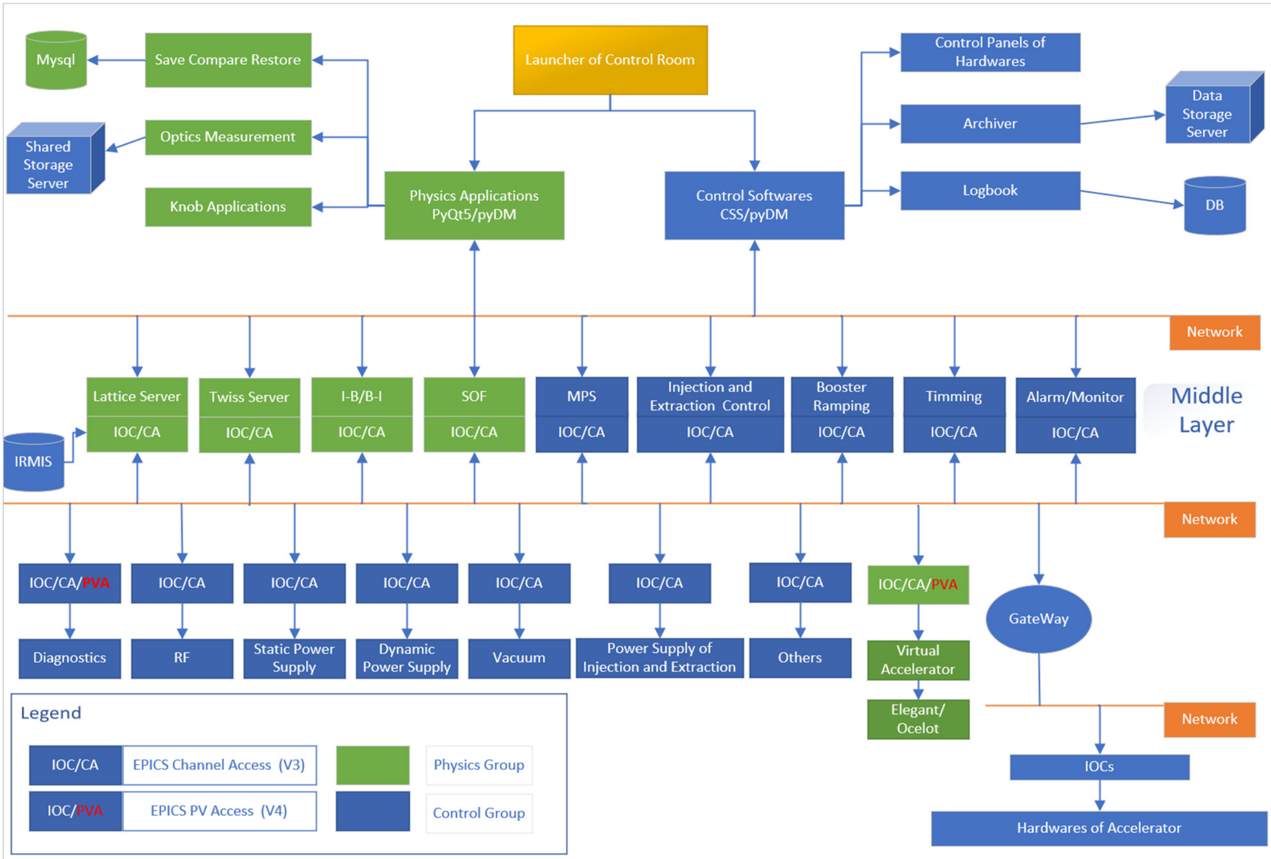

Figure 1: The structure of the control system.

make the development of model-based application intuitive, a new structure will be developed to describe the whole accelerator. For example there will be a magnet class to describe a real magnet, and an instance of this class will hold all the information of one magnet include channels and connector to call tracking algorithm. With these classes it is convenient to control machine and setup simulation, even control the machine based on simulation results. And the channel class based on EPICS for communication with machine should also be embedded. The adaptor for connecting to database is also on the development list.

The above development list describes the core elements of Pyapas, but it's not the ultimate goal. Machine learning is also on the development list, data acquisition and model training modules will be added to Pyapas. With trained model it will be easily to predict and control the actual machine through Pyapas.

To achieve the above goals, lots of development tasks have been started and some were already done. Extended QtDesigner was added to Pyapas as GUI development tool. Many customized widgets were embedded for rapid development, such as plot widgets, table widget with filter. The connectors were developed to connect to elegant and ocelot used as the core physical model. RPC and mDNS were implemented to build service applications, and adaptor for MySQL was also added to connect the database.

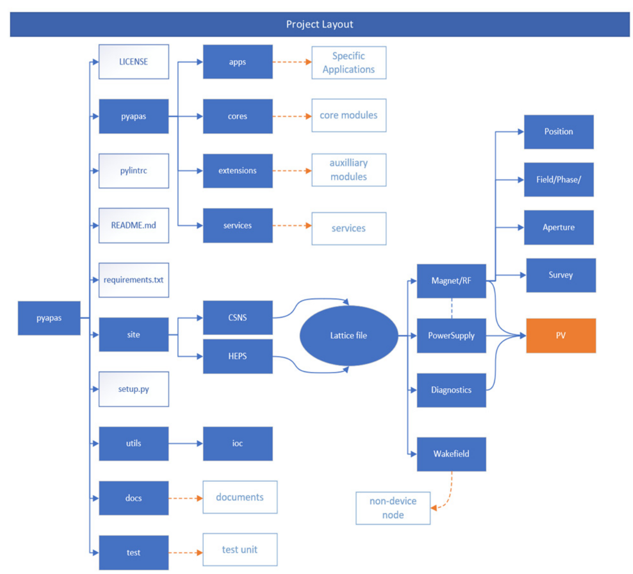

Figure 2: The structure of Pyapas.

APPLICATIONS UNDER DEVELOPMENT

For the requirements of the upcoming LINAC commissioning, the necessary applications are already under development, includes launcher, orbit correction, phase scan, emittance measurement, beam based alignment, energy and energy spread measurement, virtual accelerator and so on.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV047

As shown in Fig. 3, the launcher application is service-based. All the applications of Pyapas will register itself on the local network through mDNS, and the launcher could monitor the status of all the launched applications and do the close and launch actions.

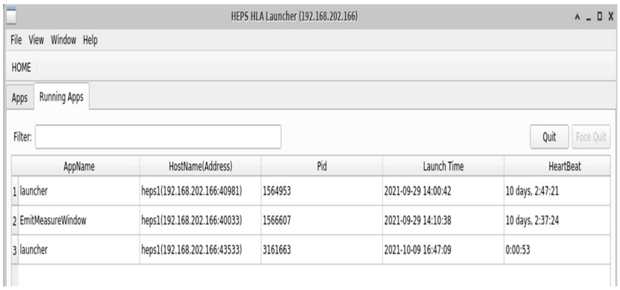

Figure 3: The snapshot of launcher.

The virtual accelerator shown in Fig. 4 is also service-based. There is a virtual IOC system which hold the same records of the real machine but start with prefix 'V'. The calculation engine built based on elegant and ocelot will supply the simulation values to the virtual IOC. For the commissioning software, the VA is same as the real machine, together with some auxiliary modules, we could also get the Twiss parameters in real time.

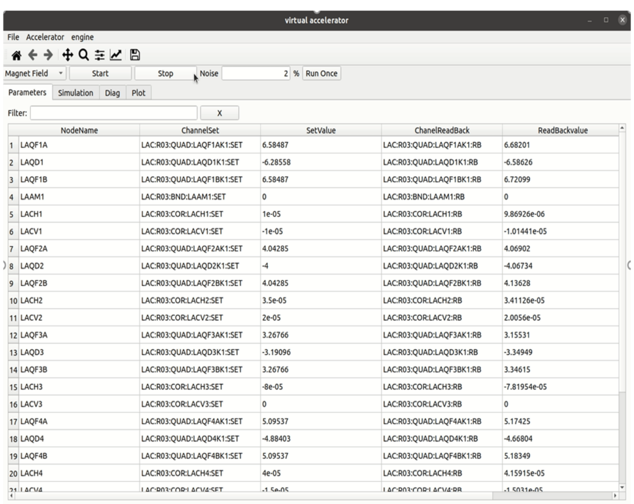

Figure 4: The snapshot of virtual accelerator.

CONCLUSION

The development of HLAs for HPES started this year. A new python-based flexible framework Pyapas is proposed to build HLA of HEPS. Pyapas is positioned as a platform for rapid development of beam commissioning software, it will be as convenient as CSS. Different from CSS for building display and hardware control applications, Pyapas is designed to build model-based applications. We keep adding new features and modules to it to make it more powerful.

To meet the requirements of upcoming beam commissioning of LINAC, the necessary applications are under development.

REFERENCES

[1] Y. Jiao, G. Xu, X. H. Cui *et al.*, "The HEPS project," J. Synchrotron Radiat. 25, 1611(2018).

[2] Y. Jiao *et al.*, "Modification and optimization of the storage ring lattice of the High Energy Photon Source", Radiat. Detect. Technol. Methods 4, 415–424, 2020. doi:10.1007/s41605-020-00189-7

[3] C. Meng *et al.*, "Physics design of the HEPS LINAC", Radiat. Detect. Technol. Methods 4, 497–506, 2020. doi:10.1007/s41605-020-00205-w

[4] Y.M. Peng *et al.*, "Design of the HEPS booster lattice", Radiat. Detect. Technol. Methods 4, 425–432, 2020. doi:10.1007/s41605-020-00202-z

[5] Y.Y. Guo *et al.*, "The transfer line design for the HEPS project", Radiat. Detect. Technol. Methods 4, 440–447, 2020. doi:10.1007/s41605-020-00209-6

[6] Y. Jiao *et al.*, "Progress of lattice design and physics studies on the High Energy Photon Source", in *Proc. IPAC'21*, Campinas, SP, Brazil, May 2021, pp. 229-232. doi: 10.18429/JACoW-IPAC2021-MOPAB053

[7] G. Portmann, J. Corbett, A. Terebilo, "Middle Layer Software Manual for Accelerator Physics," LSAP- 302, 2005.

[8] J. Corbett, A. Terebilo, G. Portmann, "Accelerator Control Middle Layer," in *Proc. PAC 2003*, Portland, OR, USA, May 2003, pp. 2369-2371.

[9] L. Yang *et al.*, "The Design of NSLS-II High Level Physics Applications,", in *Proc. ICALEPCS2013*, San Francisco, CA, USA, Oct. 2013, paper TUPPC130, pp. 890-892.

[10] I. Stevani *et al.*, "High Level Applications for Sirius," in *Proc. PCaPAC2016*, Campinas, Brazil, Oct. 2016, paper WEPOPRPO22, pp. 47-49. doi:10.18429/JACoW-PCaPAC2016-WEPOPRPO22

[11] M. Borland, "elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation", APS LS-287, 2000.

[12] I. Agapov *et al.*, "OCELOT: a software framework for synchrotron light source and FEL studies", Nucl. Instr. Meth. A, vol. 768, p. 151, 2014.

[13] Control System Studio, cs-studio.sourceforge.net

[14] PyDM - Python Display Manager, https://slaclab.github.io/pydm/

[15] A. P. Zhukov *et al.*, "Open XAL Status report 2019", in *Proc. IPAC2019*, Melbourne, Australia, May 2019, paper WEPTS096, pp. 3341-3344. doi: 10.18429/JACoW-IPAC2019-WEPTS096