PyCMSXiO: an external interface to script treatment plans for the Elekta® CMS XiO treatment planning system

To cite this article: Aitang Xing *et al* 2014 *J. Phys.: Conf. Ser.* **489** 012063

View the article online for updates and enhancements.

PyCMSXiO: an external interface to script treatment plans for the Elekta® CMS XiO treatment planning system

Aitang XING[1], Sankar ARUMUGAM[1], Lois HOLLOWAY[1,2,3], Gary GOOZEE[1,4]

[1]Liverpool and Macarthur cancer therapy centres and Ingham Institute, Liverpool Hospital, NSW, Sydney, Australia.
[2]Institute of Medical Physics, School of Physics, University of Sydney, Sydney, NSW, Australia.
[3]Centre for Medical Radiation Physics, University of Wollongong, Wollongong, NSW, Australia.
[4]South Western Sydney Clinical School, University of New South Wales, Sydney, NSW, Australia

aitang.xing@sswahs.nsw.gov.au

Abstract. Scripting in radiotherapy treatment planning systems not only simplifies routine planning tasks but can also be used for clinical research. Treatment planning scripting can only be utilized in a system that has a built-in scripting interface. Among the commercially available treatment planning systems, Pinnacle (Philips) and Raystation (Raysearch Lab.) have inherent scripting functionality. CMS XiO (Elekta) is a widely used treatment planning system in radiotherapy centres around the world, but it does not have an interface that allows the user to script radiotherapy plans. In this study an external scripting interface, PyCMSXiO, was developed for XiO using the Python programming language. The interface was implemented as a python package/library using a modern object-oriented programming methodology. The package was organized as a hierarchy of different classes (objects). Each class (object) corresponds to a plan object such as the beam of a clinical radiotherapy plan. The interface of classes was implemented as object functions. Scripting in XiO using PyCMSXiO is comparable with Pinnacle scripting. This scripting package has been used in several research projects including commissioning of a beam model, independent three-dimensional dose verification for IMRT plans and a setup-uncertainty study. Ease of use and high-level functions provided in the package achieve a useful research tool. It was released as an open-source tool that may benefit the medical physics community.

1. Introduction

Computer technology has dramatically reshaped radiotherapy in the last three decades. Dedicated software and hardware have been developed and extensively used in every aspect starting from CT image acquisition and reconstruction, radiotherapy plan simulation and dose calculations and extending to delivery incorporating record and verification systems and computer-controlled radiotherapy equipment such as linear accelerators (Linacs). For external beam radiotherapy, simulation of radiotherapy plans and dose calculation is performed with a specialized software tool called a radiotherapy treatment planning system (RTP). The Radiotherapist/Medical physicist uses the

RTP to find the optimal plan that delivers the prescribed dose to the target while sparing organs at risk. For each patient the user performs contouring of the tumor and other organs on a CT slice, adjusts the treatment and dose calculation parameters calculates dose and evaluates plan quality and then further adjusts parameters as necessary. Some or all of these tasks can be recorded or written as a series of commands in a file. Once this file is loaded into the TPS, the tasks recorded in the file can be performed automatically. Performing the tasks in the radiotherapy planning system in this way is known as treatment plan scripting [1].

Scripting can only be used in a TPS that provides an internal scripting interface. The Pinnacle (Philips) treatment planning system is one system that has inherent scripting capabilities. Its versatile scripting functionality was explored in a notable online document that contains entire collections of Pinnacle script files for various clinical tasks with a comprehensive user guide [1]. Scripting in Pinnacle has been used routinely in many centers through simple record-and-then-replay hot scripts. Advanced use of its scripting capability has also been reported for checking the integrity of treatment plans [2] as well as automation of plan generation and comparison for intensity modulated radiotherapy (IMRT) and volume modulated radiotherapy (VMAT) [3]. Another treatment planning system, Raystation from RaySearach Laboratories, also provides a built-in research interface for the user to directly interact with the core algorithms used in the system. Elekta CMS XiO is a treatment planning system that is widely used in radiotherapy centres around the world and in our centre along with the Pinnacle system. Although XiO has a non-advertised command line control scripting process, it is a non-flexible scripting interface and not recommended to be used by customers. In this work, an external interface, PyCMSXiO, was developed to enable scripting of radiotherapy plans within XiO.

2. Materials and Methods

2.1. Development of the package

The external scripting interface, PyCMSXiO, was developed using the Python programming language. Python has gained growing popularity in medical physics communities. In comparison with other programming languages, it is easy to use and has built-in advanced data structures such as list and dictionary. Because of these features, it has been used to develop several open-source packages and applications for medical physics, such as pydicom, a python package for reading and writing radiotherapy DICOM files [4], dicompyler [5], an extensible open source radiation therapy research platform and QATrack+, a radiotherapy quality assurance management system [6].

The interface was implemented as a Python package using an object-oriented programming (OOP) technique. The syntax for OOP programming in Python is simple and understandable so that one can rapidly develop a complicated software system or package that realizes design goals and can enable easy maintenance once developed. The package was well structured and contains several well-organized maintainable classes (objects).

2.2. Use of this package

As a reusable python package/library, PyCMSXiO is a tool created to enable the user to write an application to manipulate radiotherapy plans in XiO; this enables a broad range of common functions to be reused to save time and effort. The library contains common routines at a high level that allow the user to directly access and modify plan related objects. In the current version of the package, the dose recalculation within XiO for scripted plans can be triggered via the XiO user interface.

The principle and methodology for using PyCMSXiO to script XiO are exactly the same as those used for scripting in the Pinnacle system. The only difference between them is that Pinnacle scripting

uses the internal interface built into the system by the manufacturer while XiO scripting utilizes an interface external to the system. In order to script the plan objects in Pinnacle and XiO, one has to first establish a class of plan object. The plan objects can then be accessed through either the internal or external interface. This OOP scripting method is not only used for simple scripting tasks but also for complex ones usually required for a research project. The use of this package was validated for a number of research projects in our centre. For all of these projects initial plans were validated by comparing intended and resulting plan parameters.

3. Results and Discussion

3.1. Structure of the package

The PyCMSXiO was organized as a family of classes (objects). The object-oriented hierarchy of the library class is shown in figure 1 as a Unified Modelling Language (UML) diagram. There are four logical groups of classes and each is indicated by the shaded rectangular area at the top of diagram. In terms of the class relationship, Beam (object), XIOTel (object), XIOPlan (object) and Segment (object) are independent entries in the library, parallel to each other. IMRTBeam (object) and CRTBeam (object) are the children of Beam (object). These two classes inherit the properties and actions from their parents. Each class corresponds to an object in a radiotherapy plan.
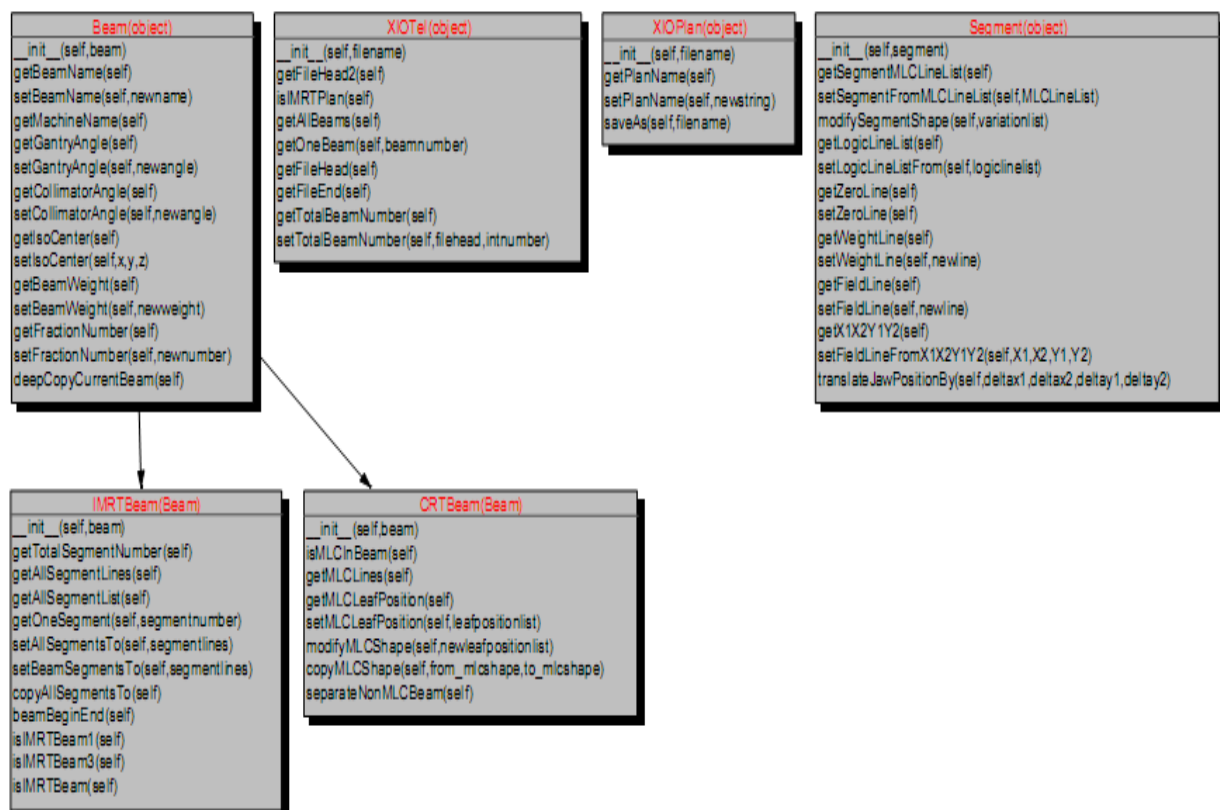
Figure 1. The hierarchy of classes (objects) in the PyCMSXiO python package.

The role of each class is to provide an interface that allows the user to interact with the corresponding object. Beam is a common object in the external radiotherapy plan. Through beam class, one can set or get beam parameters via a class function or class property. As shown in figure 1, there are thirteen object functions in its interface and these can be used to copy the beam, set or get the machine name, collimator and gantry angle, isocenter and weight point. As IMRTBeam (object) and CRTBeam (object) are the subclasses of Beam (object), they also can use the interfaces from beam object except for their own interfaces.

3.2. Comparison between XiO and Pinnacle scripting

Figure 2 demonstrates how to use PyCMSXiO for scripting in XiO and equivalent Pinnacle scripting using its internal interface. In this example, the task was to set the name, collimator and gantry angle for the current chosen beam. This is presented for scripting for both XiO and Pinnacle systems. The Beam object was established using the first- line command and the remaining lines are used to manipulate the beam parameters via the interfaces of the Beam class. The scripting syntax is slightly different between XiO and Pinnacle. This is because the Pinnacle built-in scripting interfaces are implemented as the property of class while PyCMSXiO uses object functions as its class interface. Except for this small syntax difference, XiO scripting with PyCMSXiO is comparable with Pinnacle scripting using its internal scripting interface in term of ease of use and scripting capability.

```
Beam=TrialList.Current.BeamList.Current
Beam.Name="Sup"
Beam.Gatry=270
Beam.Collimator=180
```

(a)

```
Beam=Beam(BeamList[0])
Beam.SetBeamName("Sup")
Beam.SetGantryAngle(270)
Beam.SetCollimatorAngle(180)
```

(b)

Figure 2. Comparison between (b) XiO scripting with PyCMSXiO and (a) Pinnacle scripting using its internal interface.

3.3. Application of the package

The initial motivation for developing PyCMSXiO came from a research project that required manipulation of treatment plan objects in the XiO system. Since development, the package has been used in several research projects in our department including beam model commissioning [7], independent 3D dose verification for IMRT plan [8] and patient setup studies [9]. Verification of plans generated in the initial stages of each of these projects using the PyCMSXIO package demonstrated accurate generation of treatment plan changes as expected. The use of this package is not limited to the aspects explored in our centre it could be utilised for many other research areas.

Current experience with the PyCMSXIO package has demonstrated that the ease of use and high-level functions encapsulated in this package can help users to quickly develop a tool to meet their specific needs for different research projects. Another advantage of using PyCMSXiO for scripting in XiO is that there is limited risk of harming the core XiO system as the package is an external interface and the scripting is through the Python programming language.

3.4. Package availability and limitations

PyCMSXiO was developed using free programming languages and standard libraries developed by expert volunteers from the Python community. The authors would like to distribute it as an open source tool that may be useful and benefit the medical physics community. The source code and a simple user guide are available on request. The package was only tested and used in XiO version 4.4 and is released under GNU GPL licences without any warranty.

4. Conclusion

An open-source Python package was developed for the Elekta CMS XiO treatment planning system. The package adds scripting capability to XiO through an external interface and enables the user to script radiotherapy plans generated in XiO. It has been used in several research projects and proven to be a useful research tool.

References

[1]   Geoghegan S, Gooze G et.al. *Scripting on Pinnalce 3 treatment planning system.* 2007 (www.medphysfiles.com).
[2]   Yang D and Moore K. *Med. Phys.* 2012 **39** 1542
[3]   Janssen T, Kesteren Z V et.al.   *Int J Radiation Oncol Biol Phys.* 2013 85 873.
[4]   Mason D. *Pydicom user guide.* 2013   http://code.google.com/p/pydicom/.
[5]   Panchal A and Keyes R. *Med. Phys.* 2010 **37** 3245.
[6]   Taylor R. *QATrack+: a tool for managing QA program of a radiotherapy department.* 2013 (http://randlet.com/projects/qatrack/).
[7]   Xing A, Arumugam S, Holloway L and Gooze G. *Validation of an IMRT beam model using a secondary treatment planning system as a 3D dosimeter. J. Phys.: Conf. Ser.* **444** 012051
[8] Arumugam S, Xing A , Gooze G and Lolloway L. *Independent calculation based verification of IMRT plans using a 3D dose calculation engine. Medical dosimetry.* 2013 **13** 00058-7.
[9] Xing A, Holloway L, Kumar S et.al. *The effect of set-up error on delivered dose in curative radiotherapy for Non-Small Cell Lung Cancer .* 14th World Conference on Lung Cancer (WCLC 2011) Amsterdam RAI, Amsterdam, The Netherlands, July 3 - 7, 2011.