Original Article

PMP

Progress in Medical Physics 28(3), September 2017
https://doi.org/10.14316/pmp.2017.28.3.122
pISSN 2508-4445, eISSN 2508-4453

Institutional Applications of Eclipse Scripting Programming Interface to Clinical Workflows in Radiation Oncology

Hojin Kim, Jungwon Kwak, Chiyoung Jeong, Byungchul Cho

Department of Radiation Oncology, Asan Medical Center, University of Ulsan College of Medicine, Seoul, Korea

Received 22 September 2017
Revised 29 September 2017
Accepted 30 September 2017

Corresponding author
Byungchul Cho
(bccho@amc.seoul.kr)
Tel: 82-2-3010-4437
Fax: 82-2-3010-6950

Eclipse Scripting Application Programming Interface (ESAPI) was devised to enhance the efficiency in such treatment related workflows as contouring, treatment planning, plan quality measure, and data-mining by communicating with the treatment planning system (TPS). It is provided in the form of C# programming based toolbox, which could be modified to fit into the clinical applications. The Scripting program, however, does not offer all potential functionalities that the users intend to develop. The shortcomings can be overcome by combining the Scripting programming with user-executable program on Windows or Linux. The executed program has greater freedom in implementation, which could strengthen the ability and availability of the Scripting on the clinical applications. This work shows the use of the Scripting programming throughout the simple modification of the given toolbox. Besides, it presents the implementation of combining both Scripting and user-executed programming based on MATLAB, applied to automated dynamic MLC wedge and FIF treatment planning procedure for promoting the planning efficiency.

Keywords: Eclipse Scripting API, Data-mining, Plan quality metrics, Dynamic MLC wedge, Field-in-Field

Introduction

Radiation treatment is to deliver the dose of radiation to a patient throughout a lot of procedures ranging from delineating target volume and structures to treatment planning, dose delivery and quality assurance. Each step usually demands a series of the labor-driven, time consuming tasks. Also, facing the era of big data, to acquire and review a huge number of patients with similar tumor locations and prescriptions, it requires for accessing a large amount of data each at a time from the limited information. The recent significant developments in radiation oncology, therefore, have been made to lessen the manual interventions and human labors in treatment associated applications by utilizing high computing device and algorithms e.g. GPU,[1] and deep learning.[2]

Keeping in pace with the advances, several vendors providing treatment planning system (TPS) have made an effort to help users to improve the working efficiency for treatment, and offered a Scripting application programming interface (API). This is a toolbox based on C# programming, which is designed to conduct a variety of functionalities. It ultimately pursues attaining automation in a variety of applications in radiation therapy, such as the treatment planning, file transmission, data mining and verification of plan quality. A couple of studies[3,4] have shown that the scripting can be used in automated treatment plan optimization and data-mining. Besides that, the Scripting API could be employed for far more applications throughout simple modifications.

The Scripting program, nevertheless, could not cover entire potential applications that the users intend to do. The functions provided by Scripting API are limited in applications if the users want to process a couple of dicom information. Instead, it provides the environment where a user-executed program could be run in combination with file import/export system. The Scripting program automatically exports/imports the dicom files from TPS, and enabling for running an executed file that was created by the users in Windows and/or Linux OS. With the framework, the users are able to comprehend a number of functions in the executed program. As being implemented in Wang et al.[5] combining the user executable program would broaden the usage of the Scripting API in the clinical applications.

This work first presents a series of clinical applications of the Scripting API provided by Varian Eclipse (Varian Medical Systems, Palo Alto, USA) TPS, called Eclipse Scripting API (ESAPI). In addition to the simple modifications of the ESAPI, it shows that the concept of combining the user-executed program with the ESAPI is able to be used for automated treatment planning in dynamic MLC wedge[6,7] and Field-in-Field (FIF)[8-11] techniques.

Materials and Methods

1. Framework and variation of scripting API

Fig. 1 illustrates a structure of the Scripting API provided by Varian Eclipse, which was based on C# programming language. The toolbox grants a couple of exemplified codes, thus allowing the users to run those for clinical applications. Those codes are designed to perform when the user has access to each patient data. For instance, if the plan quality needs to be quantified throughout the Scripting program, which will be shown in the next section, the patient data should be open to implement the specific script code on the TPS. It results in the patient-specific consequences, the quantified plan quality according to the above example.

There exist the other cases, where the user wants to investigate clinical information of many patients. Under the circumstances, it has to reach out to the database of TPS such that it looks over all listed patients. The Eclipse Scripting API offers the functionality in the form of stand-alone data-mining program. It is possible for the user to put a list of patient IDs or names, then getting the stand-alone program run over the scheduled computations after having the user to log into the system. This would let the system yield the results on the command prompt, while it could be set up to create a text or excel file with resulting values, depending on the applications.

The other possible approach described in Fig. 1 is to utilize the user-executable program. The C# code modifications in the Scripting API is useful in some

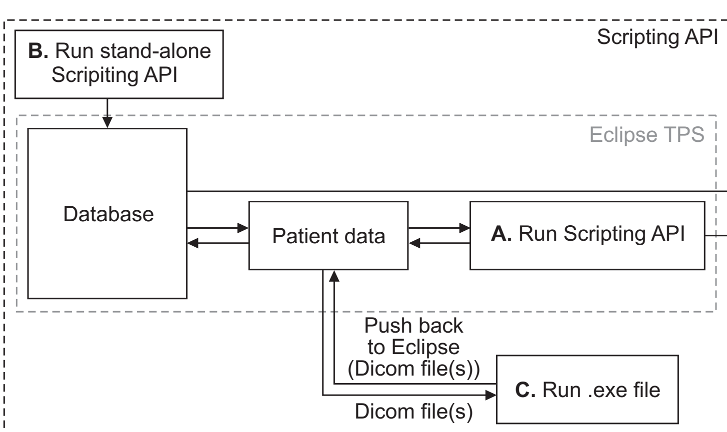

Fig. 1. Framework of Scripting API communicating with Eclipse TPS. It runs the API in three different ways: A. Running on each patient dataset, B. Accessing DB with stand-alone program, and C. Implementing an execution file created by users to conduct the intended tasks.

applications, while it does not cover whole probable applications that the users intend to conduct. Once the tasks are associated with managing the dicom header information, it is likely that the Scripting program does not allow for its compatibility to the system for clinical safety. The shortcomings could be complemented by creating the programs that the users desire to perform. It would make a Window- or Linux-based program created, and run on the Scripting programming by connecting an execution (.exe) file to the TPS in Fig. 1. Any functions can be developed in the user-created programs that fit to the clinical application. Wang et al.[5] made a user-executable program on C# programming language, while the execution files in this work were created in MATLAB programming. Once the user generated the execution file for a certain application, the Scripting program provides an environment that helps run the execution file. In addition to it, given that the user created program requires for a couple of the dose, structure, image, and plan dicom files to be run, the Scripting program in Eclipse also offers the ability of file export/import between the server and the designated local directory.

2. Applications of scripting API

The combination of the user-executed program on Windows with the Scripting API is of interest as it is expected to extend the territory of the Scripting API once clinically permitted. In the Results section, this study shows a couple of the examples from the combination, e.g. automated dynamic wedge and field-in-field (FIF) techniques. They mostly focus on the MLC and/or MU adjustments without manual interventions to promote the automation and efficiency in treatment planning. Besides, without combining the executable file, the Scripting API code provided run by itself for the clinical applications as stated above. We would show an example of the given Scripting API throughout the simple code modifications to plan quality metrics, and the usage of the stand-alone programming for the data-mining purpose.

Results

1. Plan quality metrics

The function about plan quality metrics is given by the Eclipse Scripting API. It could be used to confirm the plan based on the numerical information of the dosimetric results, which is to see if the resulting dose distribution meets the required dose constraints for the target volume and critical structures. For the given code, it is possible for the users to edit the list of dose constraints. We have added a constraint for the target volume to specify the target dose conformality, called conformality index (CI) or conformation number (CN).[12-14] It consists of the two terms: the dose coverage that was already included in the constraints, and the spilling factor that quantifies the high dose of radiation exposed to the healthy tissue around the target, as described in Eq (1):

$$CI = \frac{V_{\tau,ref}}{V_{\tau}} \times \frac{V_{\tau,ref}}{V_{ref}} \quad (1)$$

where V_{τ} is the delineated planning target volume. And $V_{\tau,ref}$ and V_{ref} are the target and the entire volume, respectively, receiving the dose greater than the prescribed dose. Depending on the study, the equation in (1) is reciprocally defined. Fig. 2 shows a report created after running the Scripting code regarding plan quality metrics, where the dotted box in red indicates the conformity index added to grant more specified planning information.

2. Data mining

The example above is the result obtained from running the Scripting API for each patient data. There, however, might be many cases where the users have to access the database of TPS, such that a number of patient datasets can be explored at a time. The data-mining function given by the Scripting API enables for having access to the DB of the Eclipse TPS by logging into the system as seen in Fig. 3. Once getting into the system, it conducts the designated tasks. The example in Fig. 3 is to search for the patients in DB, and measure the volume of structures contoured in different imaging modalities, as being computed in the

Structure information

Target structure: PTV_NG

Volume : 66.31 cc

Name	Type	Evaluation Point	Calculated Value	Goal	Must	Evaluation
PTV coverage (V[95%(Rx)>95%])	VolumeAtDose	2,200.0 cGy	95.033%	95.000%	90.000%	PASS

Structure information

Target structure: PTV_NG

Volume : 66.31 cc

Name	Type	Evaluation Point	Calculated Value	Goal	Must	Evaluation
PTV coverage (V[95%(Rx)>95%])	VolumeAtDose	2,200.0 cGy	95.033%	95.000%	90.000%	PASS
Spillage (V_PTV[100%(Rx)]/V[100%(Rx)])	Spillage factor	2,200.0 cGy	88.2%	80.0%	75.0%	PASS
CI=PTV coverage*Spillage	Conformity Index	2,200.0 cGy	83.8%	75.0%	70.0%	PASS

Fig. 2. Plan quality metrics by modifying C# code of the Scripting API to add the conformality index of the target volume (dotted box).

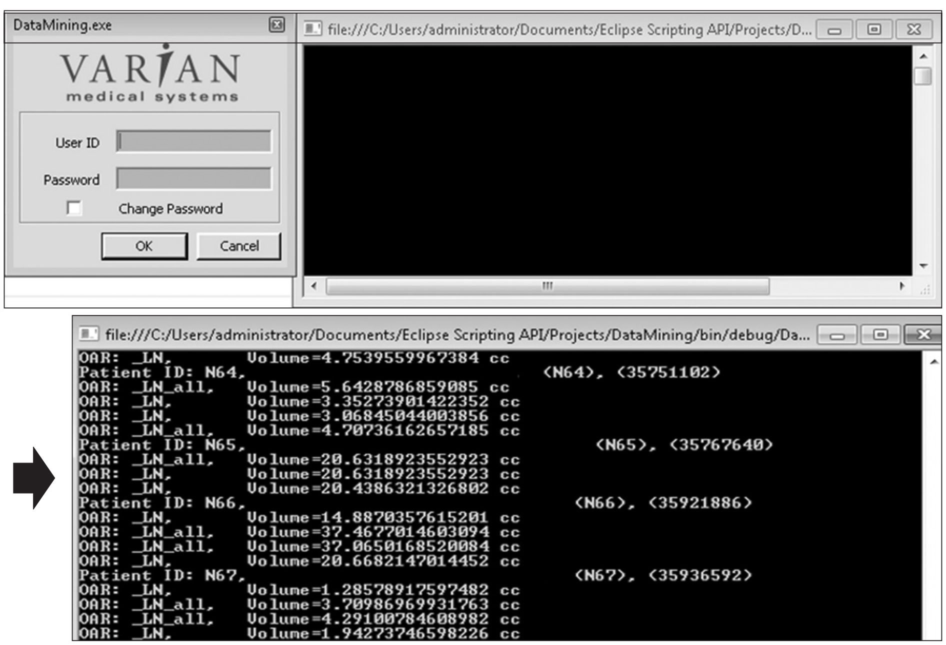

Fig. 3. Example of data-mining throughout the Scripting API, which was designed to measure the volume of the designated organs contoured on different imaging modalities.

command prompt. The result could be written in text and/or Excel format in the local directory relying on the user preference.

3. Combining user-executed programming for treatment planning

The Scripting programming gives an environment, where the user-executed programming runs. As described in Fig. 1, it transmits the patient dicom files to a specified path, which will be used for the user-created program to conduct the scheduled tasks. The resulting dicom file after run is pushed back to the TPS throughout the Scripting as well. The combination of the file import/export function from Scripting, and user-executed program would be more powerful as the tasks that the users want to implement get too complicated to be processed by the Scripting programming itself. Our institution has been

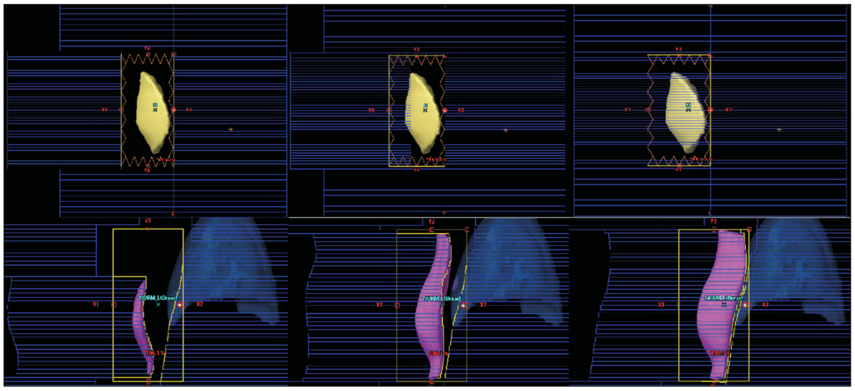

Fig. 4. Automated MLC-pair adjustment for two-types of planning procedures throughout a combination of Scripting programming with user-created execution program (Top: Dynamic wedge, Bottom: Field-in-Field (FIF)).

trying to adopt the combination to enhance the efficiency in treatment planning procedure by automating MLC and/or MU adjustment in 3D forward plans. Specifically, the concept was employed for automating dynamic wedge based planning, and field-in-field (FIF) techniques, as illustrated in Fig. 4.

The top row of Fig. 4 shows the automatic wedge shaping with dynamic MLC movement in lateral direction, which could replace the use of physical wedge. The bottom row describes the automated FIF forward IMRT technique that adds the sub-fields by adjusting the MLC shapes to the segmented dose clouds with assigning appropriate MUs. The user-executed programming mainly governs the MLC movement, such that it can go through the in-field region at a certain speed for the dynamic wedge application, and block the segmented dose clouds by adjusting MLC pairs with appropriate MU settings, respectively. The details of the applications will be specified in the currently reviewed articles.

Discussion

The Scripting API has been developed to promote the efficiency of clinical applications in radiation oncology, therefore achieving automation in whole procedure. It is a toolbox made of C# programming language, compatible to the Eclipse TPS system. The toolbox provided a couple of exemplified codes for the users to take it to such applications as data-mining, plan quality metrics, automated treatment planning, and efficient quality assurance procedure. As the codes are available, it is possible for users to modify it for the other specific cases for both clinical and research purposes. As seen in the preceding section, we could add the clinically useful dose constraint, conformality index, to the given constraints. Also, we have changed the searching condition and the measurements in data-mining application. These examples demonstrate that the toolbox does not only improve the efficiency, but also be able to attain better safety.

The Scripting toolbox gets more powerful by combining its ability with the user-executable programming in Windows or Linux. The Scripting API may restrict the flexibility of the operations for clinical safety or other purposes. By the combination, there would be far greater potential applications available in radiation oncology. Our institution has been employing the idea for enhancing the efficiency in treatment planning. The executable program was designed to automatically adjust the MLC shapes and/or MU weights for dynamic MLC wedge and field-in-field techniques. As discussed in Results section, MLC pairs are well controlled fitting to the segmented dose clouds, which do not compromise the plan quality relative to the plans manually done. Of note, implementing the automated dynamic wedge and FIF planning with a combination of the Scripting and user-created executing programs takes about 30 seconds for conducting a series of auto-segmenting dose clouds, projecting segmented dose to beam shape, and adjusting MLC/MU, which may need a few more minutes for refining the MLC pairs if necessary. It is a significant enhancement in planning efficiency, relative to the manual task that elapses at least more than 10~15 minutes for the procedure.

We have been trying to use the Scripting program based frameworks actively for both clinical and research purposes. The data-mining tool is being used for research mainly in collecting and looking into the patient datasets. In particular, the automated FIF plans are being applied to 10 to 20 cases for breast cancer treatment planning every week. The notion of combining Scripting with the user-executable program is considered very powerful, so it could be employed for numerous applications especially when the users need to modify the dicom files and plan parameter settings. However, this workflow must be carefully formulated, such that it firmly observes the clinical regulations. In most cases, the users would have to confirm the tasks by consulting with oncologists and manufacturers before taking it to the clinical applications.

Conclusion

This study presents a series of possible clinical applications of the programming toolbox, the Scripting API provided by Eclipse TPS. The toolbox is used for various cases by modifying the programming codes, while its ability could be more strengthened if it is combined with user-created program on Windows or Linux. We have shown the examples of its applicability in plan quality metrics, data-mining, and user-executed programming aided, automatic MLC dynamic wedge/FIF planning. When applying those to our institutional workflows, it has been dramatically improving the efficiency as well as the safety.

Acknowledgements

This work was supported by the National Research Foundation of Korea grant funded by the Korea government (NRF-2013M2A2A7043506).

Conflicts of Interest

The authors have nothing to disclose.

Availability of Data and Materials

All relevant data are within the paper and its Supporting Information files.

Ethics Approval and Consent to Participate

The study was approved by the institutional review board (IRB approval number; 2016-0321).

References

1. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Computation 2006;18(7):1527-54.
2. Jia X, Ziegenhein P, and Jiang SB. GPU-based high-performance computing for radiation therapy. Phys Med Biol. 2014;59(4):R151.
3. Xhaferllari I, Wong E, Bzdusek K, Lock M, Chen J. Automated IMRT planning with regional optimization using planning scripts. J Appl Clin Med Phys. 2013;14(1): 4052.
4. Hill P, Ramalingam E. Data mining with eclipse scripting. Physica Medica. 2016;32(7):955.
5. Wang H, Xing L. Application programming in C# environment with recorded user software interactions and its application in autopilot of VMAT/IMRT treatment planning. J Appl Clin Med Phys. 2016;17:189-203.
6. Zhu, J. Generation of wedge-shaped dose distributions through dynamic multileaf collimator dose delivery. J Appl Clin Med Phys. 2005;6(3):37-45.
7. Mayo C, Lo YC, Fitzgerald, TJ and Urie, M. Forward planned, multiple-segment, tangential fields with concomitant boost in the treatment of breast cancer. Med Dosim. 2004;29(4):265-70.
8. De la Torre, N., Figueroa, C.T., Martinez, K., Riley, S. Chapman, J. A comparative study of surface dose and dose distribution for intact breast following irradiation with fieldin-field technique vs. the use of conventional wedges. Med Dosim. 2004;29(2):109-14.
9. Borghero, Y.O., Salehpour, M., McNeese, M.D., Stovall, M., Smith, S.A., Johnson, J., Perkins, G.H,, Strom, E.A., Oh, J.L., Kirsner, S.M., Woodward, W.A., Yu, T.K. Buchholz, T.A. Multileaf field-in-field forward-planned intensitymodulated dose compensation for whole-breast irradiation is associated with reduced contralateral breast dose: a phantom model comparison. Radiother Oncol. 2007;82(3):324-8.
10. Prabhakar R, Julka PK, Rath GK. Can field-in-field

technique replace wedge filter in radiotherapy treatment planning: a comparative analysis in various treatment sites. Australas Phys Eng Sci Med. 2008;31:317-24.
11. Morganti AG et al. “Forward planned intensity modulated radiotherapy (IMRT) for whole breast postoperative radiotherapy. Is it useful? When?,” J Appl Clin Med Phys. 2011;12(2):213-22.
12. A. van’t Riet, A. C. Mak, M. A. Moerland, L. H. Elders, W. van der Zee. A conformation number to quantify the degree of conformality in brachytherapy and external beam irradiation: Application to the prostate,” Int. J. Radiat. Oncol., Biol., Phys. 1997;37(3):731-6.
13. R. Oozeer, B. Chauvet, R. Garcia, C. Berger, C. Felix-Faure, F. Reboul. Dosimetric evaluation of conformal radiotherapy: Conformity factor, Cancer Radiother. 2000; 4(3):207-15.
14. L. Feuvret, G. Noel, J. Mazeron, P. Bey. Conformity index: A review,Int. J. Radiat. Oncol., Biol., Phys. 2006; 64(2):333-42.