Calculated Contour Generator (CC-Gen)

Dominic J DiCostanzo
Department of Radiation Oncology
The Ohio State University
Columbus, OH 43210
dicostanzo.2@osu.edu

Abstract

Purpose: To develop a script that would allow for the generation of calculated structures from the anatomic and target segmentations to support the increased utilization of knowledge-based planning models.

Methods: An application extension was developed that utilizes a text file input to support different planning techniques and styles. The input file describes the operations necessary to create planning organ-at-risk volumes or optimization structures as well as the ID and DICOM type to be applied. The native treatment planning system operations were implemented and extended.

Results: The software has been in clinical use since 2019 and is expected to have over 2800 uses at the end of 2023. It is now being used by both dosimetrists during planning and radiation oncologists for margin creation. The increase in use of the software is correlated with an increase in the use of knowledge-based planning models, Pearson correlation coefficient of 0.87.

Conclusion: The generation of calculated structures via an automated process saves treatment planners time, decreases the likelihood of random errors during structure creation, and has positively influenced the use of knowledge-based planning at our institution.

***K*eywords** automation · segmentation · knowledge-based planning · scripting · ESAPI · ROI algebra

1 Introduction

Commercially introduced in 2013, knowledge-based planning (KBP) has shown promise in creating radiation therapy treatment plans that are non-inferior to manually generated treatment plans.[1] One barrier to use is the user of the model must segment anatomy and create planning structures (segmentations) similar to those which were used in the model training.[2] Since KBP utilizes the geometrical relationship of segmentations to predict the dose volume histogram (DVH), segmentations that are dissimilar to those used to train the model will result in suboptimal plan generation due to model extrapolation and necessitate manual planning. Even with detailed descriptions of the structures and the steps to generate them, users unfamiliar with their purpose will be challenged in reproducing the required structures. Reducing this barrier is paramount to wider adoption and translation of KBP models.

Segmentation is a central task of modern radiation therapy treatment planning. Most, if not all, commercial treatment planning systems (TPS) include modules enabling users to generate segmentations in support of radiotherapy treatment plan creation. In addition to TPS, other commercial software solutions exist which support segmentation via manual and automated methods. Automated segmentation methods commonly implemented range from atlas-based to machine and deep learning.[3] Regardless of the method used to create anatomic or target segmentations, calculated structures are often required to create or evaluate treatment plans. Unlike anatomic segmentations, calculated (or derived) structures refer to those volumes of interest (VOIs) created in support of radiation therapy treatment planning that are based on anatomic segmentations.

In addition to being dependent on the anatomy of individual patients, these calculated structures can be institution, TPS, disease site, and planner dependent. Their generation can take considerable time per patient. Moreover, if a patient undergoes offline adaptive therapy whereby a new computed tomography (CT) simulation is acquired, the anatomic segmentations, and any dependent calculated structures, must be recreated to support the generation of a new treatment plan. While some TPS include tools for regenerating calculated structures if its base segmentation changes, this is not common.

Prevalent among modern TPS is the implementation of an application programming interface (API) that allow users to create custom software to extend the system in various ways. The Eclipse® TPS (Varian Medical Systems, Palo Alto, CA) includes an API referred to as the Eclipse Scripting API (ESAPI). ESAPI enables developers to read data from and write data to the TPS. These input/output (I/O) operations open new possibilities for increased efficiencies and increased safety. One such possibility is the development of a script for the generation of calculated structures, both PRVs and optimization structures.

The purpose of this work was to develop a script that would allow for the generation of calculated structures from the anatomic and target segmentations, however created (i.e., manually, or automatically), to support the increased utilization of KBP models.

2 Methods

ESAPI version 15.1, C#, and Visual Studio (Microsoft Corp., Redmond, WA) were used to develop the structure generator software. A common programming practice, modular programming, was utilized whereby the larger program was divided into smaller sections to support easier debugging and future extension. As such, two separate programs were created during the development of this software. First, a class library that contained the methods to support the user-facing software was created (i.e., a helper library). A text file input was selected as the user interface to provide a simple method of creating and modifying multiple user templates. The Calculated Contour Generator (CC-Gen) is dependent on the helper library, i.e., references it. CC-Gen was developed as a plug-in to enable the end-user to generate contours and review them without saving; allowing for quick evaluation of templates and the ability to undo scripted contour creation if an unexpected result was achieved.

The development of the helper library began with the implementation of the native actions supported in the TPS. Some of the actions added to the library include the Boolean operations (e.g., and, or, not) as well as subtraction and basic cropping. Each action was then extended to support different C# classes including collections (e.g., lists or enumerables). As each action was extended, it was tested to ensure the accuracy of the scripted segmentation versus a segmentation manually generated using the same actions.

2.1 User Interface

CC-Gen was programmed to accept a text file whereby each line is a segmentation to create or overwrite and the logic to create it. An equal sign was used to separate the structure identification from the generation logic. [See Figure 1]. The logic for structure generation starts with the selection of an initial structure that must exist in the structure set (base structure). For some operations the base structure is ignored and can be entered as the "Body," but for others the base structure acts as the initial segmentation that all operations are applied. Multiple operations can be chained through the use of a period. This delimiter was chosen as it is a common programming construct called dot notation. For each operation requested, any arguments that may be needed (e.g., margin size in millimeters) are supplied in parentheses, to mirror another common programming construct, the calling of a function or method.

2.2 Input Syntax

Chained operations and parenthetical arguments were utilized as the syntax of the text files to streamline segmentation creation. Chained operations are applied in the order they are entered, and a user can apply any number of operations for the creation of a structure. For example, if the line in a text file was entered as: _Avoid, CONTROL = SpinalCord_05.Or(BrainStem_03).AsymMargin(0,50,0,0,0,0). A new structure with ID of "_Avoid" of DICOM Type "CONTROL" would be created. The segmentation would combine spinal cord PRV and brainstem PRV and then add a 50mm posterior margin. In this case, the chained operations are the Or and AsymMargin operations.

If a structure with the ID requested already exists, a user can enter a tilde ( ) or exclamation point (!) as a prefix to the line, omit the DICOM Type, and the software will overwrite the segmentation with matching ID. For instance, ∼_Avoid = SpinalCord_05.Or(BrainStem_03).AsymMargin(0,50,0,0,0,0), would repeat the previous example, but the result would overwrite the segmentation already present in the _Avoid structure.

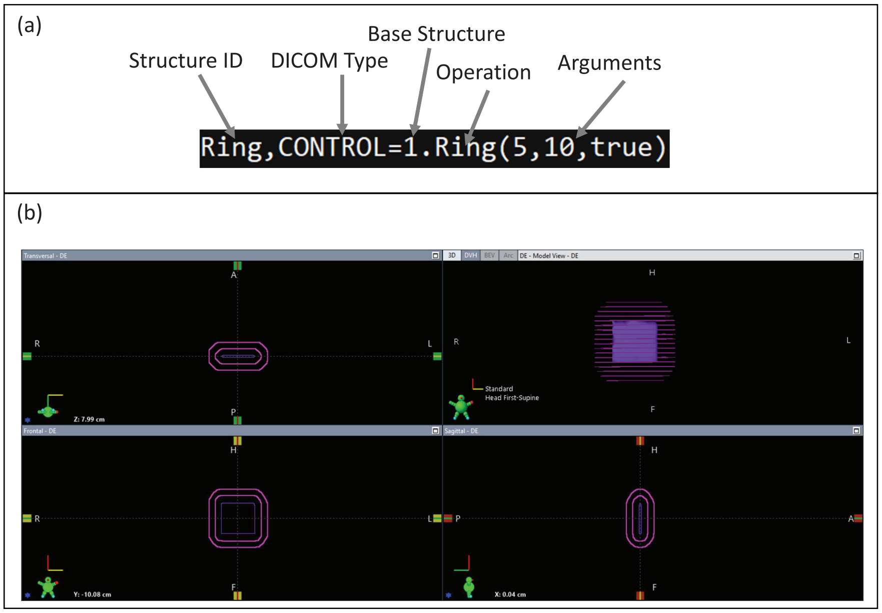

Figure 1: (a) A visual diagram of the text input. When creating a new structure in the structure set, the left side of the statement (separated by an equal sign) contains the structure ID and DICOM type separated by a comma. The right side contains a base structure with which the operations are applied. The operations are preceded by a period and any arguments for the operation are contained within the parenthesis. (b) The resultant structure created by the statement in (a). A high-resolution ring that begins 5mm from the base structure (ID of “1”) and is 5mm in width ending at 10mm from the base structure.

2.3 Operations and Arguments

The software supports all of the native Boolean, cropping, and margin operations available in the TPS, with the addition of many others. The Ring operation creates a ring from the base structure and takes arguments for the start distance, end distance, and whether the resultant ring should be created as a default or high-resolution segment. The PTVALL operator creates a segmentation that is the union of all segmentations with IDs starting with the characters “PTV.” A more generic operator, ALL, was developed that accepts a search string and where to search (e.g., whether the IDs being searched starts with, contains, or ends with the search text) and then searches the entire structure set for those structures that match the provided arguments.

An iterative operator, LIST, was implemented that has similar operators as ALL, but will iteratively perform any chained operations on the segmentations that match the search parameters and iterate the resultant segmentations numerically. For example: PTV, PTV = Body.List(GTV,Starts).HighRes().Margin(3). This command would find all of the structures in the structure set that start with “GTV”, create a new structure with ID “PTV1” of type “PTV”, convert it to high resolution, and fill the empty segmentation with a 3mm margin from the first GTV found. It would then iterate over the remaining GTVs increasing the digit for each subsequent PTV created (e.g., PTV2, PTV3, PTV4, etc.). The base structure “Body” is ignored in this operation, but it must exist in the structure set.

2.4 Error Handling

In the event that the software encounters an error, the user is prompted with an error message. [See Figure 2] The operation causing the error is skipped, but the remaining lines of the template are processed. If a structure is to be overwritten but it does not exist in the structure set, the software will prompt the user with a warning but proceed to create the structure with the supplied ID and a default DICOM type of CONTROL. If a line in the text file begins with a “#” then it is treated as a comment and skipped. Blank lines are also skipped to allow for better organization of the text file.

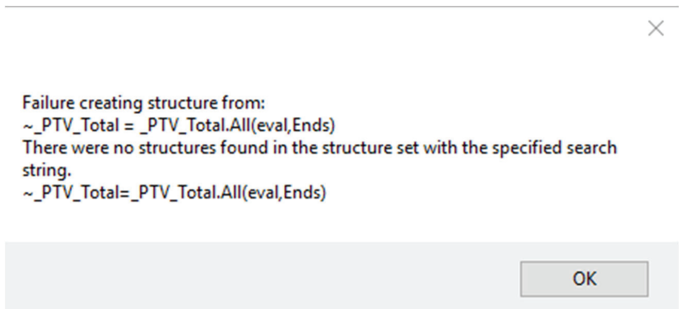

Figure 2: An example of the error message displayed to a user when an exception is encountered during the creation of a structure.

Table 1: The fit metrics and Pearson correlation coefficient for the relationship between the use of CC-Gen and associated use of KBP models.

Goodness of Fit Metrics	
Coefficient of Determination	0.75
Pearson Correlation	0.87
p-value	0.02

2.5 Commissioning and Validation

After development of the software a series of checks were performed to ensure that any segmentations created would be identical to those completed by manual processes. In addition, as new use cases were identified, the software has been extended to meet those needs. During each upgrade or modification, a similar comparative tests are performed to validate the software before it is put back into clinical use.

3 Results and Discussion

The initial version of CC-Gen was put into clinical use in September 2019 and is publicly available (https://github.com/ddicostanzo/CC-Gen). The treatment planning team was trained on the software and instructions (see GitHub repository) with example templates were published in our department. Initial uptake was slow, likely due to the learning curve associated with a new piece of software. As staff became comfortable and realized its benefits, the number of uses increased dramatically. Currently, there are over 100 different templates to be used with the software including a template that pairs with each KBP model to support its use. [See Figure 3]

The creation of templates that pair with KBP models is an important part of our KBP implementation. One of the main factors in the limited uptake of such models at our institution was the differences in planning techniques between treatment planners. This manifests as different names and geometries for optimization structures. If a planner created a KBP model and tuned the optimization objectives to the structures that they would use personally, then this created a barrier to use of the models. The creation of a CC-Gen template in support of a KBP model has reduced the barrier to use and has increased our institution's use of KBP models. [See Figure 4 and Table 1]

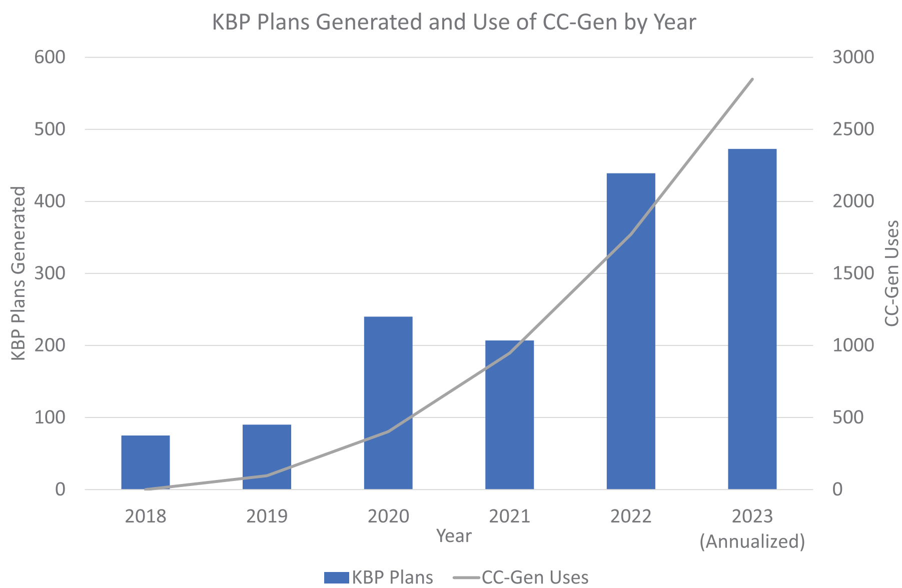

Figure 3: The increased usage of both knowledge-based planning models and CC-Gen software by year.

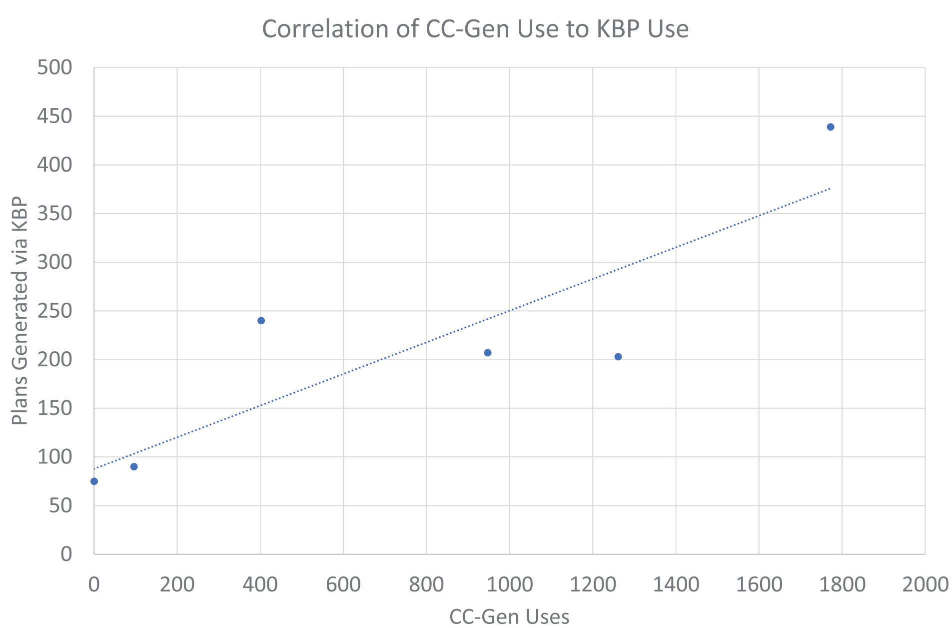

Figure 4: The correlation between increased software utilization and plans generated via knowledge-based planning models.

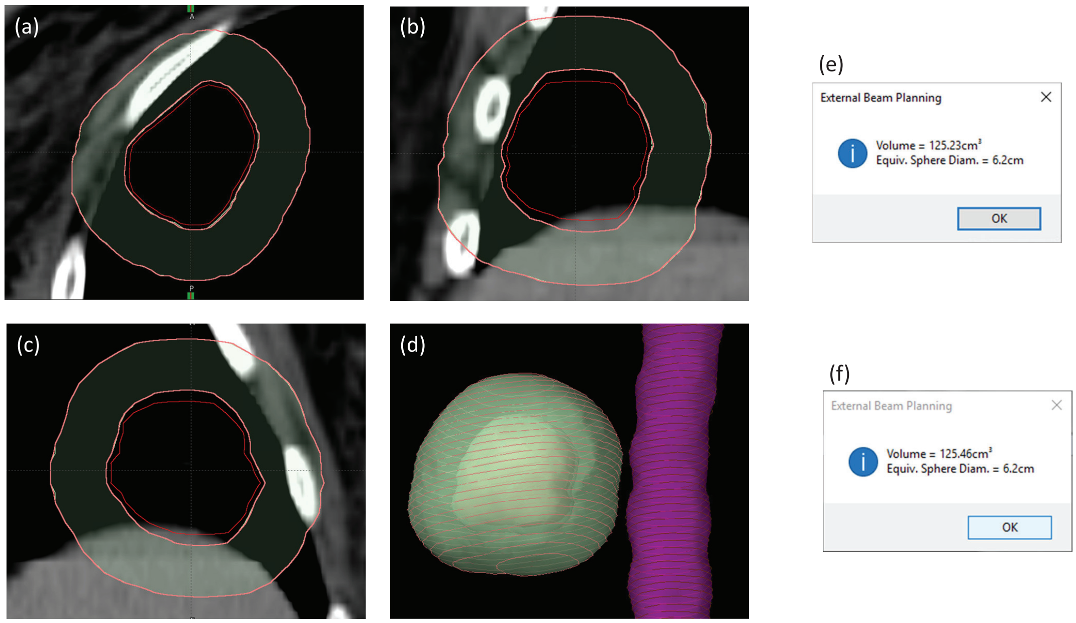

Figure 5: A visual comparison of manually generated structures and those created by the software: (a) axial view, (b) sagittal view, and (c) coronal view. A three-dimensional view (d) displays the software generated structure as a wire-frame for both a ring structure and the spinal cord with 5mm margin. A comparison of structure volume is also presented with (e) representing the manually generated structure and (f) the software created one.

As mentioned previously, each of the native segmentation operations were extended to additional C# classes, and they were also extended to support non-native conditions. For instance, margins greater than 5cm are not supported in the TPS but via the API it is possible to support the creation of margins of any size. Another example, the TPS is unable to perform actions on segmentations of different resolutions (i.e., default resolution structures and high-resolution structures, those with closer spacing of points outlining the volume of interest). Logic was created to overcome this limitation and provide the user a more seamless experience when creating calculated structures. [See Figure 5]

The CC-Gen has been in use since late 2019 and has seen increased utilization over time. The pairing of templates with KBP models has increased the utilization of both CC-Gen and KBP models in our institution. The generation of calculated structures via an automated process saves treatment planners time, decreases the likelihood of random errors during structure creation, and has positively influenced the use of KBP at our institution.

Acknowledgments

The authors would like to acknowledge Dr. Ahmet Ayan and Dr. Ashley Cetnar for feedback on the manuscript. We would also like to acknowledge our colleagues in the dosimetry team for their feedback on the software during its development and ongoing feature requests.

References

[1] Mariel Cornell, Robert Kaderka, Sebastian J Hild, Xenia J Ray, James D Murphy, Todd F Atwood, and Kevin L Moore. Noninferiority study of automated knowledge-based planning versus human-driven optimization across multiple disease sites. *International Journal of Radiation Oncology* Biology* Physics*, 106(2):430–439, 2020.

[2] Jim P Tol, Alexander R Delaney, Max Dahele, Ben J Slotman, and Wilko FAR Verbakel. Evaluation of a knowledge-based planning solution for head and neck cancer. *International Journal of Radiation Oncology* Biology* Physics*, 91(3):612–620, 2015.

[3] Carlos E Cardenas, Jinzhong Yang, Brian M Anderson, Laurence E Court, and Kristy B Brock. Advances in auto-segmentation. 29(3):185–197, 2019.