Web-App Development for Viewscreen Live Image Acquisition

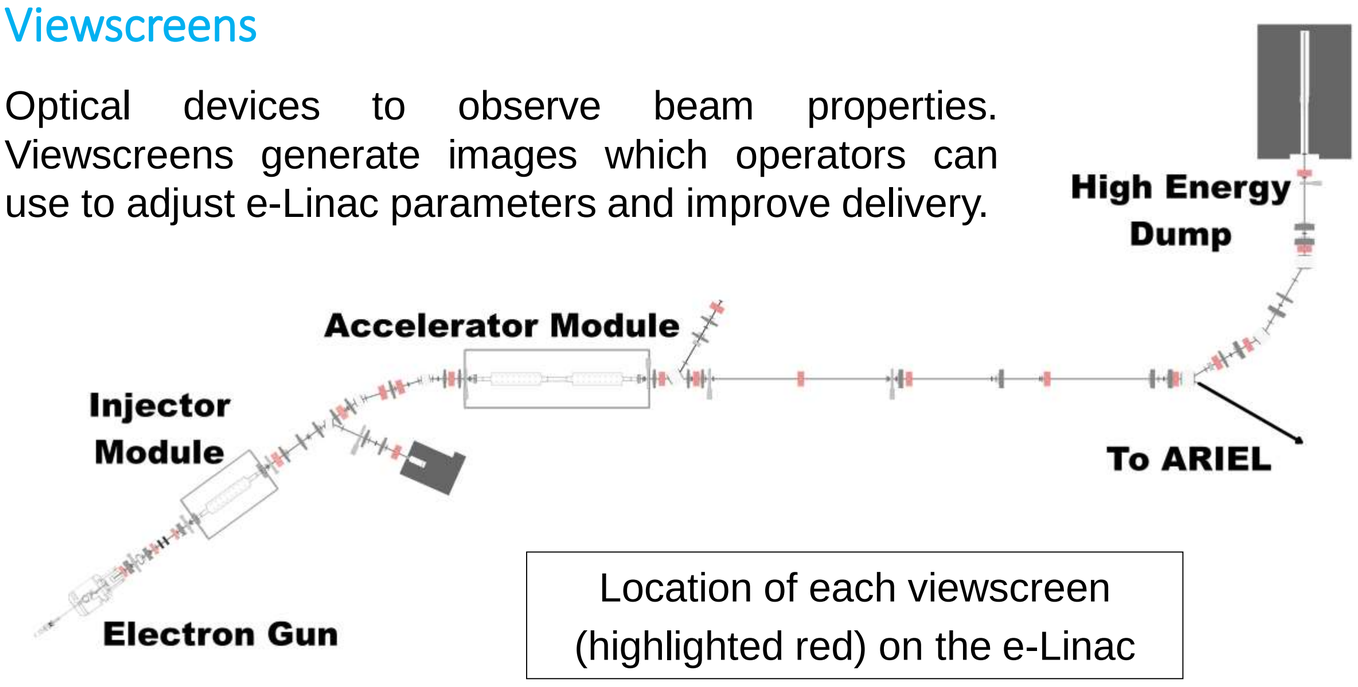
A. J. Hayes, H. W. Koay, P. M. Jung

Abstract

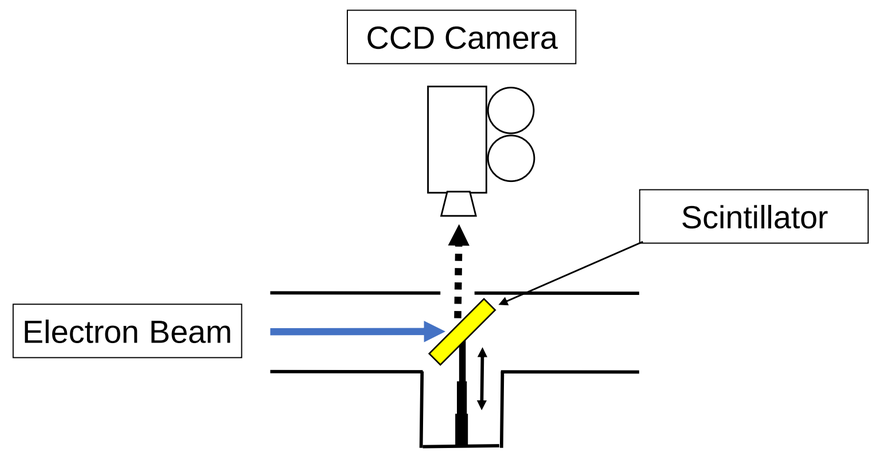
This work is to reform the user experience of operating the ARIEL e-Linac viewscreens by implementing a Python script to a centralized web-application (web-app) within the High Level Applications (HLA) framework, which can be accessed over the internet from any device. This web-app allows for click-of-a-button image capturing and automatic handling of control system parameters. The web-app also enables analysis of previously captured viewscreen images by streamlining database searching. Furthermore, the app will provide control room operators with correct beam size measurements, facilitating operations without the need for experts to be present. The usability and reliability enhancement that this web-app provides for all control room users when operating the viewscreens will significantly enhance efficiency in troubleshooting beam issues and ramping power.

Viewscreens

Optical devices to observe beam properties. Viewscreens generate images which operators can use to adjust e-Linac parameters and improve delivery.



Location of each viewscreen (highlighted red) on the e-Linac



Simplified diagram of viewscreen operation

Web-App

A simple, easy-to-use web-based interface allows for efficient location of tools and options. Images can be captured and saved to the beam database though the app at the click of a button.

Beam Database Searching

Streamlined database searching gives a complete view of recent viewscreen images, searchable by date. Each database entry contains a link to display the raw viewscreen images.

Display Page

Clicking the link brings the user to a display page: a full display of spatial beam properties alongside the raw viewscreen image. In this page, the user can download the viewscreen data or run a processing script to visually identify the beam and perform analysis, displayed to the right of the original image.

Backend Structure

This web-app has a novel software architecture; using the python package Celery, a task runner asynchronously performs the image capturing or processing, while the web server retrieves the results in real time and sends them to the user's web page. This web-app is the first HLA app to use this technique, paving the way for other apps to implement the same structure.