SIMULATION WORKSTATION

Thomas J. Roberts#, Muons, Inc., Batavia, IL 60510 USA

Abstract

The Simulation Workstation is a software toolkit that provides a universal Graphical User Interface to most particle simulation codes. This includes: constructing and displaying the simulated system graphically, running multiple simulation codes from a single system description, displaying particle tracks or histories with the objects of the system, flexibly generating plots and histograms, and comparing the results from multiple simulations. The workstation itself knows very little about particle simulations -- that knowledge is contained in the simulation codes themselves. As a consequence, the workstation can handle essentially any problem that any of the supported simulation codes can simulate. These include: beam optics calculations, ion source design, muon cooling channels, spacecraft radiation levels, nuclear reactors, complex shielding calculations, and accelerator driven subcritical reactors. The workstation offers interfaces to most CAD/CAE programs, enabling workflows that include multi-physics analysis by other programs. The Simulation Workstation will be portable among all major operating systems, will be freely available as open-source software, and will initially support these simulation codes: G4beamline [1], MCNP6 [2], and MAD-X [3]; additional codes can be added by users.

INTRODUCTION

When teaching beam dynamics and accelerator physics to students, user-friendly graphical tools are a big advantage, as they permit the construction and exploration of models with very little overhead. And for today's technology-driven students of the "Nintendo generation", graphical interfaces meet their expectations far better than text-based tools. Moreover, there are dozens of modeling codes available, so having a common user interface that can be used with multiple simulation codes will be a major advantage. Existing tools fall short of what is needed.

While the workstation has been designed with students in mind, accelerator physicists will also find it useful in dealing with the plethora of modeling tools and their different languages and data formats. The internal representation of the system is specifically designed to be useable as a text-based description of the system, and to make it easy for users to interface it to essentially any accelerator-modeling tool, regardless of its description language. In particular, this will make it straightforward to use fast but less realistic codes to design and optimize a system, and then use slower but more realistic codes to evaluate its performance. Graphical interfaces are emphasized, making it easy to construct the system graphically, display the system and its beam, and use on-screen controls to vary parameters and observe their effects immediately. Such exploration is essential to give students insight into how systems behave, and can be valuable to the experienced accelerator physicist, for tasks such as evaluating different techniques for tuning a system before it is built.

The user interface is centered around objects, which can be individual accelerator components, or arbitrarily complex collections of components with specified spatial relationships. Objects are stored in Libraries, which can easily be shared with other users. The use of URL-based component libraries will encourage collaboration among geographically diverse teams.

The Simulation Workstation provides the user with a single user interface to construct, design, optimize, analyze, and explore systems involving charged particle beams and other radiation. It is specifically designed to be interoperable with just about any existing simulation or beam optics code. This is possible for one simple reason: they necessarily have a common domain, with common concepts, common objects, and common operations. By abstracting these common elements into a new specification language, called *LinguaFranca*, a single GUI program can indeed interface to many simulation codes with relatively little effort. LinguaFranca has been specifically designed to make it easy to *instantiate* its description for other codes to read. Instantiation is not machine translation, which is known to be a difficult problem, but rather is performed by a sophisticated text-processing system that has been told by human developers how each LinguaFranca primitive is represented in the language of each simulation code.

The architecture of the Simulation Workstation is to be an overall control program that orchestrates the assembly of the various pieces required to prepare, perform, and analyze a simulation. Thus the user works with a single user interface to perform all related tasks. While it relies on interfaces to the simulation codes, the adaptation is always in the workstation, and no simulation code needs to be modified in order to work with it (some codes will benefit from modifications, especially on Windows).

The Simulation Workstation will be portable among the major operating systems (Windows, Linux, Mac OS X), and will be released as open-source software. This is the same overall commercialization model as for our highly successful program G4beamline [1]. The key to G4beamline's popularity is its excellent user interface; this project was inspired by that interface, and by what we perceived as ways to improve it. But rather than just improve this one code, we have chosen to design and develop a system that is customizable to improve the user experience for all related simulation codes.

tjrob@muonsinc.com

ISBN 978-3-95450-138-0

THE SIMULATION WORKSTATION

The Simulation Workstation presents the user with a window ready to edit a simulation, as shown in Figure 1. The viewer is at the left, showing a 3-D image of the current system; the simulation shown is for the storage ring of a Very Low Energy Neutrino Factory [4]. The right-hand third of the window shows an edit panel at the top and a library panel at the bottom (showing a single row of three objects' icons, with scroll bar).

The 3-D Viewer

As a simple illustration of the viewer's flexibility, Figure 2 zooms in on the transition from straight to arc. This required the following simple user gestures:

1. Use the middle mouse button to pan the display to center the desired region.
2. Use the mouse scroll wheel to zoom in.
3. Use the viewer-control slider to set the solid opacity to 50%.
4. Use a viewer-control button to toggle the display of axes to on.
5. Use the left mouse button to select the last cell of the straight section (highlighted in red).

Total elapsed time: 3-5 seconds. The white line is the design coordinate axis, from the origin off the screen to the left, going once around the ring, and then right (+Z) to infinity. Design coordinates make it easy to lay out an accelerator or beamline by sequentially placing objects along the system centerline; bending magnets are accompanied by a bend in the centerline; they behave just like the implicit reference coordinates of beam optics programs like MAD-X.

The Edit Panel

The edit panel in the upper right is context sensitive, displaying appropriate information in response to user gestures.

The Library Panel

The library panel at lower right displays the objects currently in the library, sorted alphabetically. These can be simple parameterized components such as magnets and RF cavities, or they can be combinations of other objects, such as :Straight and :Arc (which define a cell for the corresponding region of the VLENF). In general there can be multiple Libraries (remote ones are configured via URLs), so library components are named LibraryName:name (no spaces). This is the default library whose name is empty, so the names simply begin with a colon. Objects can be dragged into the viewer and dropped in position; the edit panel can then be used to

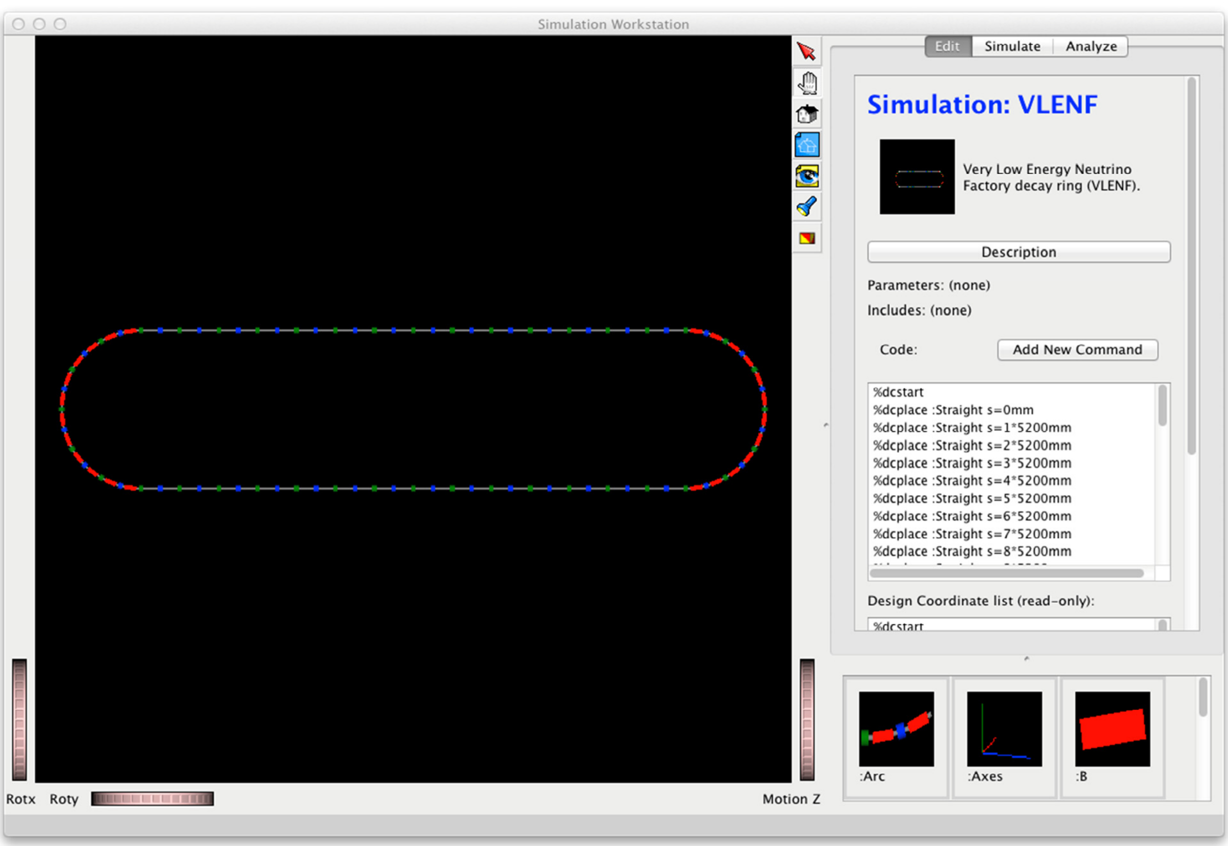

Figure 1: The Simulation Workstation Main Window, editing a VLENF simulation. Focusing and defocusing quads are green and blue, bending magnets are red, and beam pipes are gray. The 3-D viewer can be easily rotated, panned, and zoomed via the mouse, and individual objects can be selected and edited.

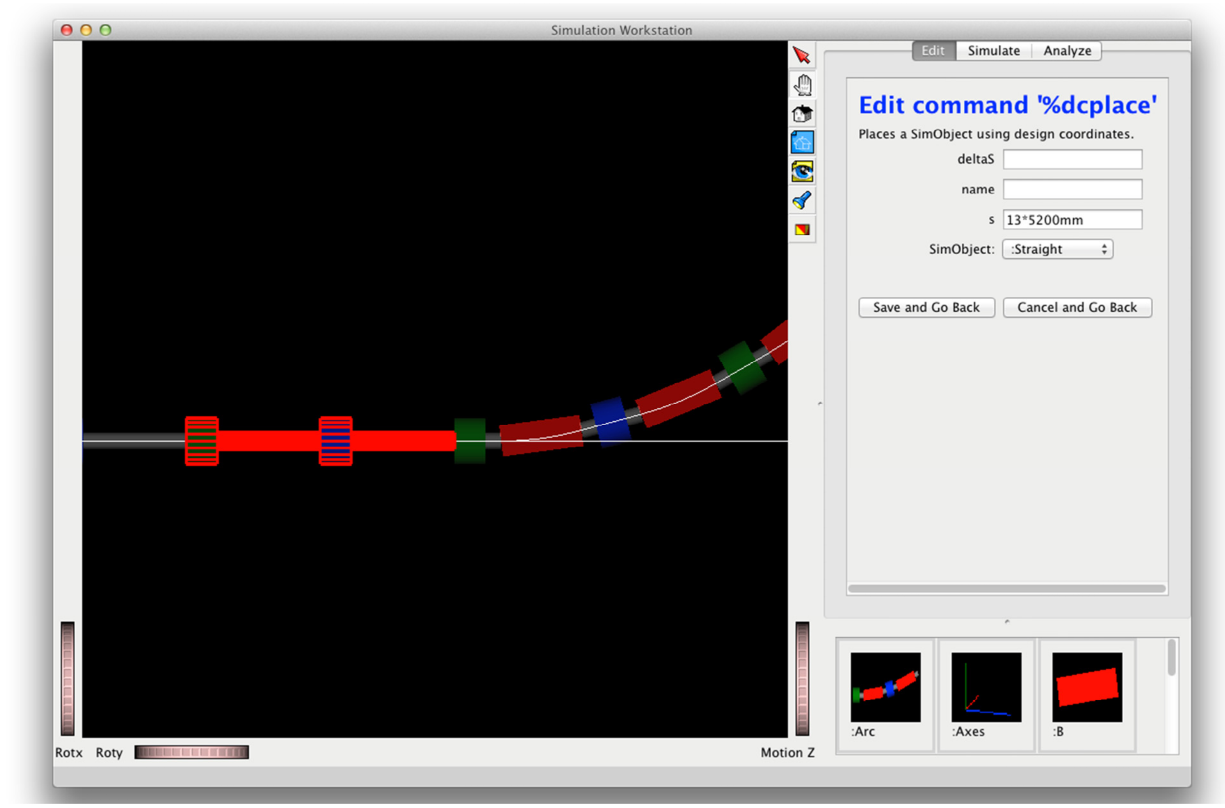

Figure 2: The system of Figure 1, but panned and zoomed, with the last Straight cell selected. Note the editing panel shows the command that put this cell into the simulation, placing the cell along the design coordinate axis.

tweak their position precisely. But for an accelerator system using Design coordinates, once the components are placed into the library they can simply be dragged into the edit panel and dropped – this places them sequentially along the design coordinate axis; The VLENF simulation has magnets and beam pipes placed sequentially into cells, and the cells placed sequentially into the simulation.

The Simulate Tab

The upper right panel has three tabs that change the entire mode of operation: Edit, Simulate, and Analyze. The Simulate tab provides an interface to manage the running of a simulation, using any configured simulation code. As a simulation runs, its output file(s) can be scanned as they are produced, and events displayed in sets of N (which defaults to 10 but is easily changed). A viewer-control slider permits the user to vary the opacity of solids, making it easy to see tracks inside them. Clicking on a track does two things: it hides any other events in the current set, and in the upper right it displays the track values at the point clicked and at track creation. By default tracks are colored by their charge, but the user can configure a color for each type of particle.

The Analyze Tab

The Analyze tab will make it easy for the user to create histograms and plots of quantities of interest. It will incorporate a complete Root [5] package, as well as an interface to simplify the creation of basic histograms and plots, with sliders to apply cuts (updating the plot in real time as the cuts are changed). The Analysis tab will incorporate feedback to the Simulate tab, permitting the user to identify interesting, unusual, or “outlier” events in a plot, and then re-simulate each event while viewing its tracks in the viewer. With G4beamline [1] this has proven to be a very powerful technique for understanding backgrounds and rare processes.

CONCLUSION

The Simulation Workstation is expected to become a useful addition to many students’ and accelerator designers’ toolkits. It has many innovative features not found in other programs that promise to improve users’ productivity; its use of libraries should foster collaboration among geographically diverse teams.

REFERENCES

[1] G4beamline http://g4beamline.muonsinc.com
[2] MCNP6 http://mcnp.lanl.gov/
[3] MAD-X http://mad.web.cern.ch/mad/
[4] VLENF http://arxiv.org/abs/1206.0294
[5] ROOT http://root.cern.ch/

ISBN 978-3-95450-138-0