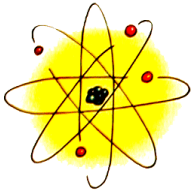




Joint ICTP-IAEA Workshop on Monte Carlo Radiation Transport and Associated Data Needs for Medical Applications
28 October – 8 November 2024
ICTP, Trieste, Italy

Lecture 14

Introduction to BEAMnrc

David W O Rogers

Carleton Laboratory for Radiotherapy Physics
Physics Department
Carleton University, Ottawa

OMEGA project and the BEAMnrc code

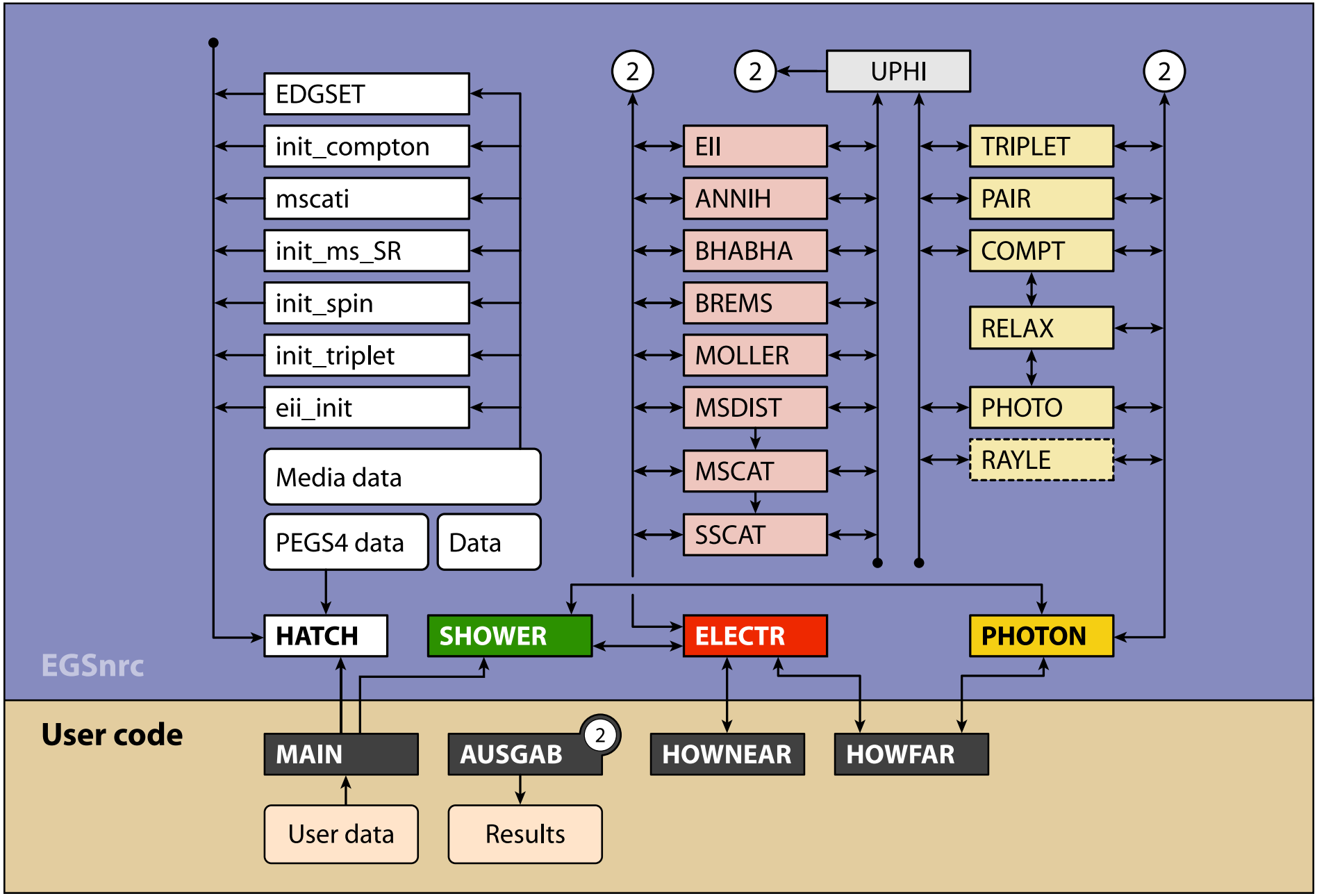
- BEAM is a general purpose code to simulate radiotherapy beams
  - accelerators: electron and photon beams
  - Co-60 units
  - x-ray units
- originally part of the OMEGA project done in collaboration with Rock Mackie's group in Madison (1990 - 1996)
- OMEGA: **O**ttawa **M**adison **E**lectron **G**amma **A**lgorithm
- many grad students, RAs and TOs involved

BEAM/BEAMnrc developers

Dave Rogers
Iwan Kawrakow
Bruce Faddegon
Charlie Ma
Michel Proulx
Joanne Treurniet
Blake Walters
Ernesto Mainegra-Hing
George Ding
Geoff Zhang
Daryoush Sheikh-Bagheri
Jiansu Wei

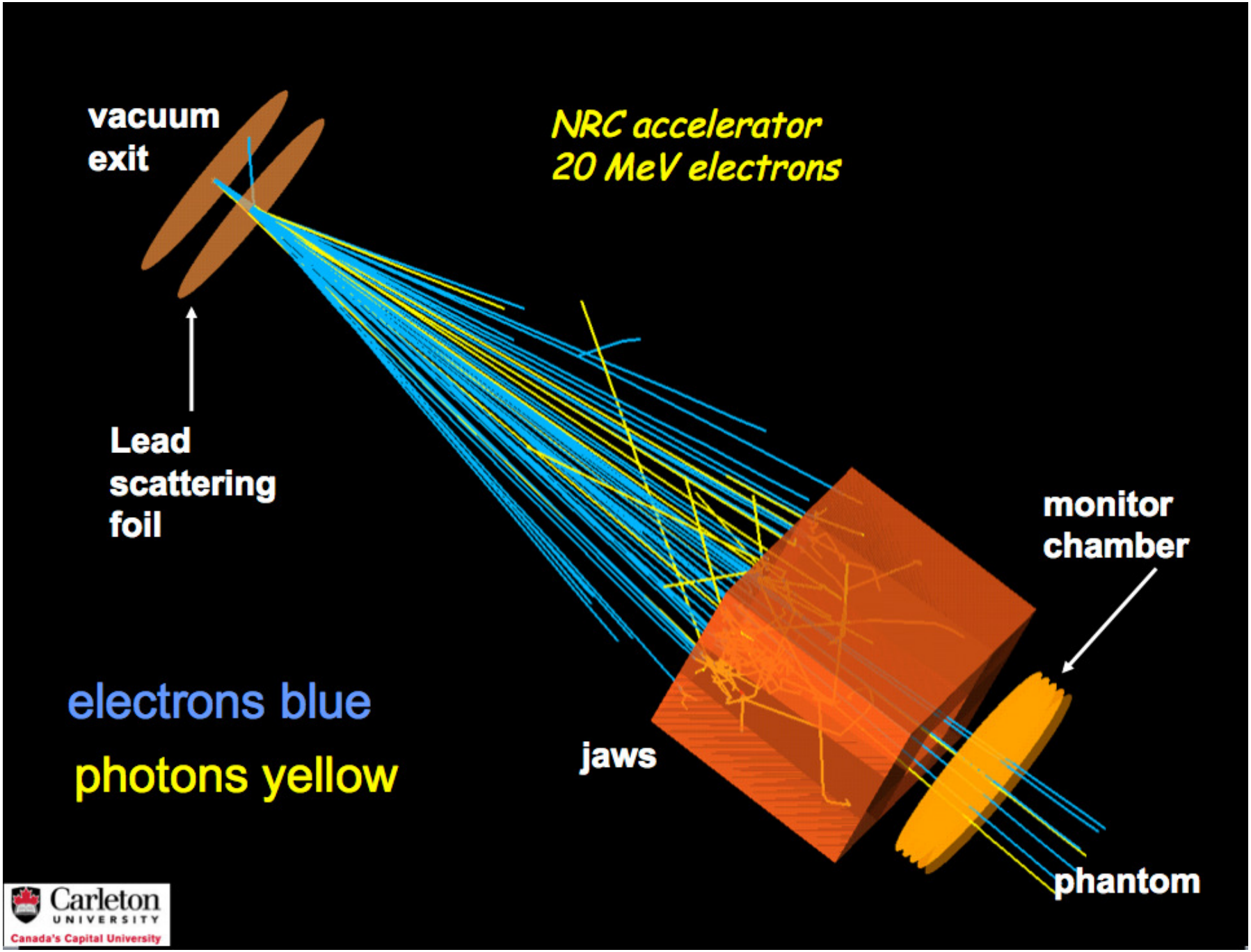
EGS4 developers: Ralph Nelson and Alex Bielajew
EGSnrc developer: Iwan Kawrakow
EGSnrc maintenance: Ernesto Mainegra-Hing, Frederic Tessier, Reid Townson, Blake Walters

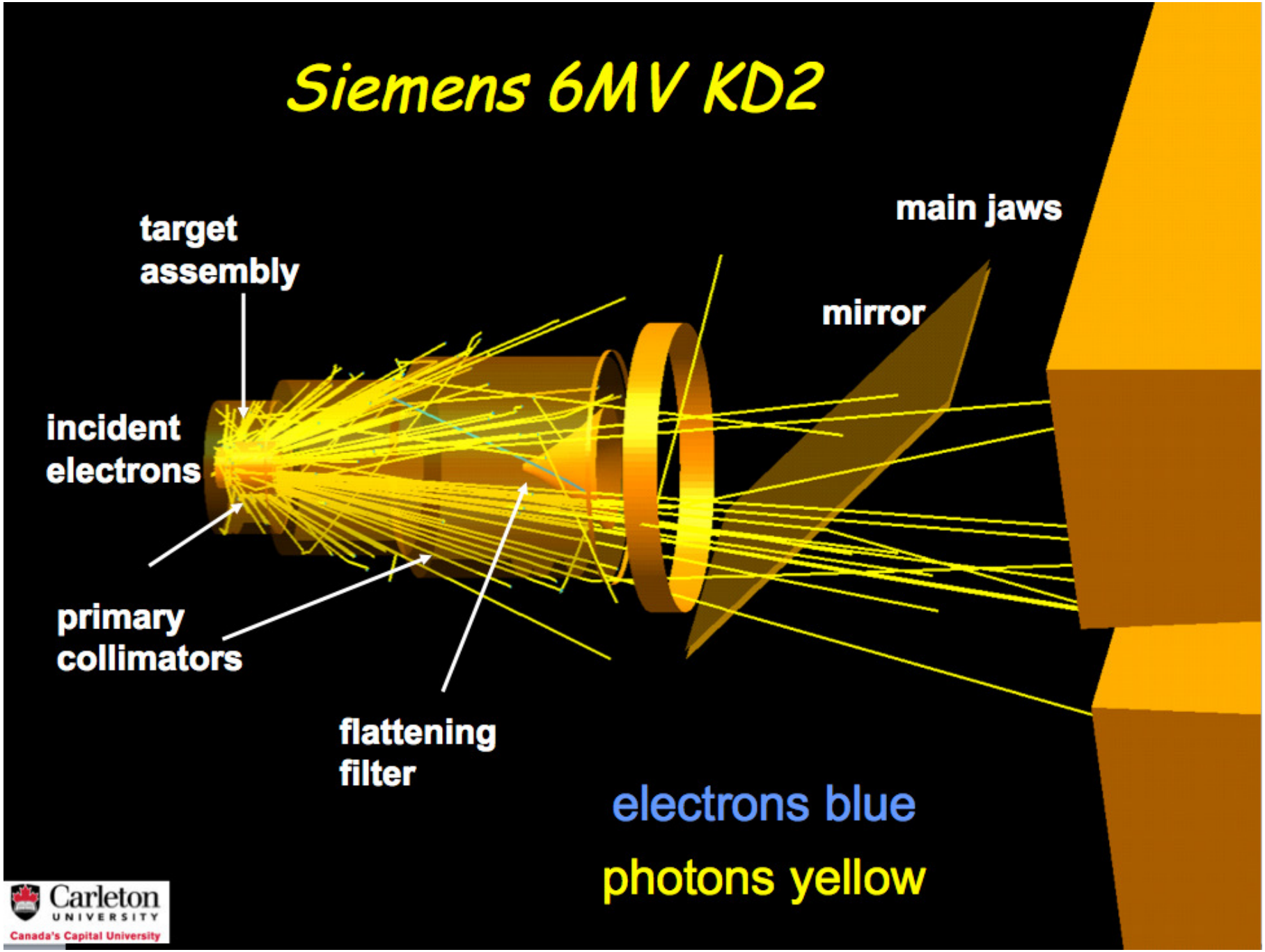
BEAMnrc is built on EGSnrc

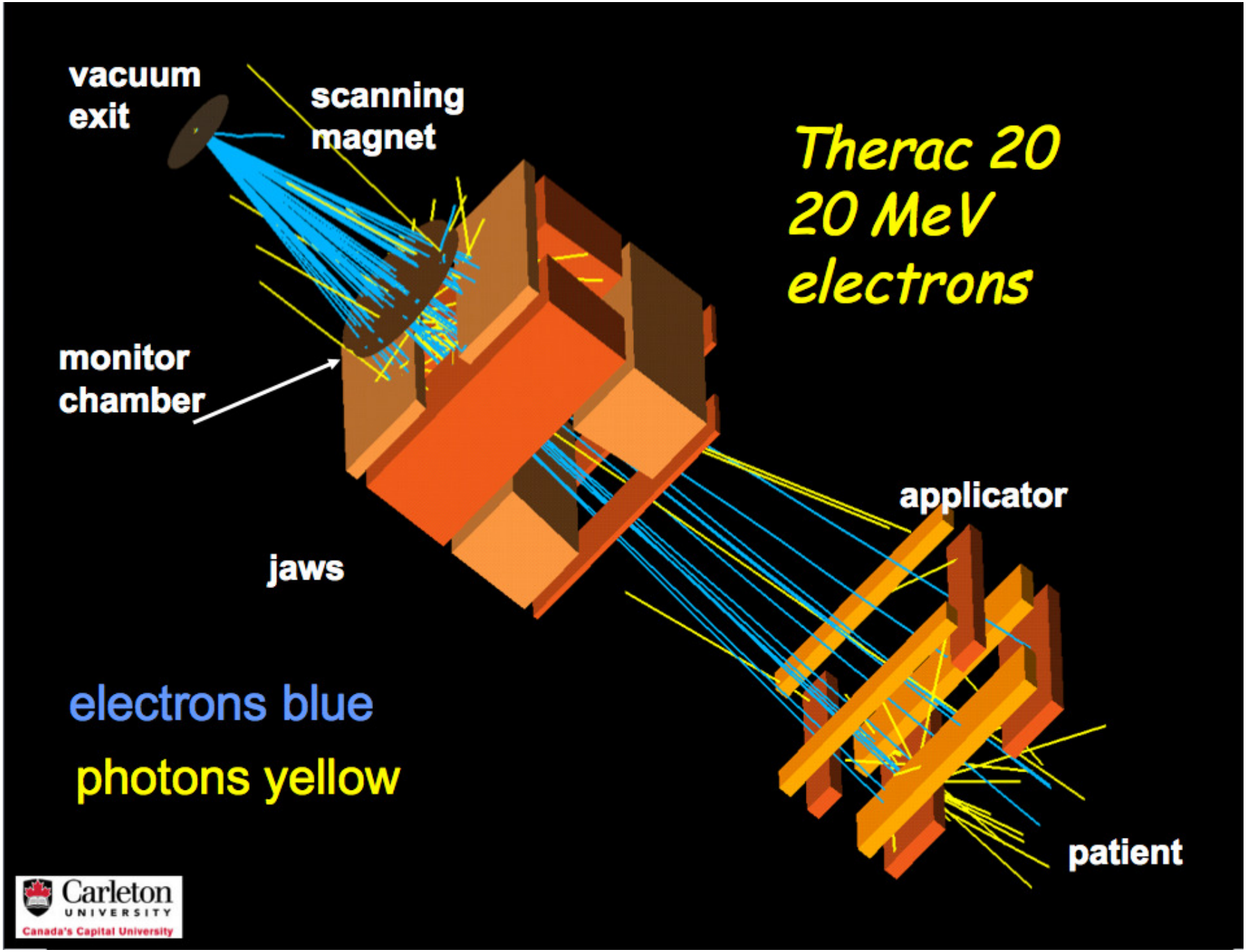


BEAMnrc design features

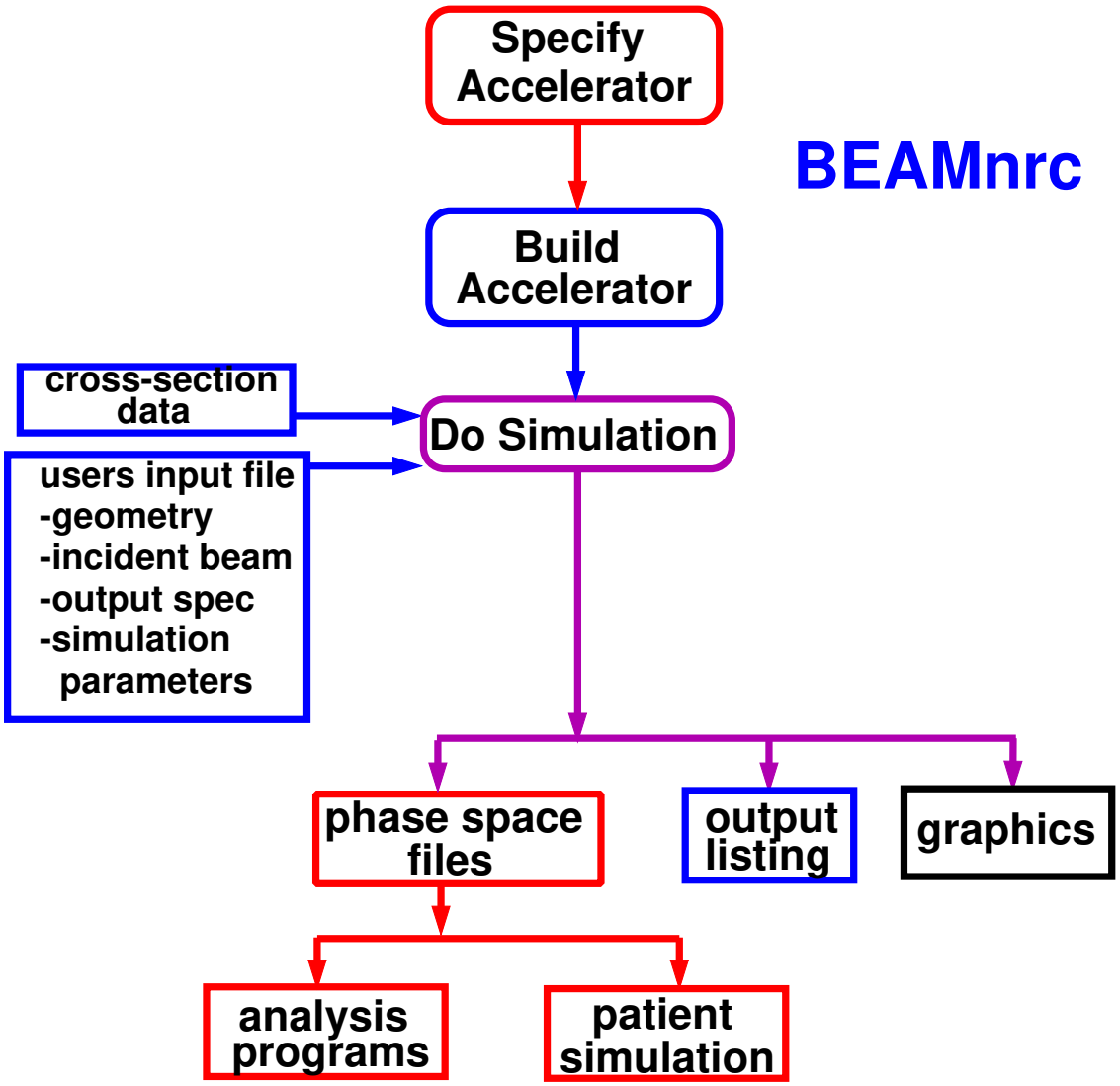
- **C**omponent **M**odules (CMs)
  - between parallel planes
  - can be combined in arbitrary order
  - builds in flexibility and extensibility
  - conceptual simplicity
- not restricted to cylindrical symmetry (all prior models were cylindrical)
- variance reduction for accelerator modelling was built in from the start
- detailed testing
- expert user friendly (need to be an expert to use well)


vacuum
exit
NRC accelerator
20 MeV electrons
Lead
scattering
foil
monitor
chamber
electrons blue
photons yellow
jaws
phantom
Carleton
UNIVERSITY
Canada's Capital University


Siemens 6MV KD2
target assembly
main jaws
mirror
incident electrons
primary collimators
flattening filter
electrons blue
photons yellow
Carleton UNIVERSITY
Canada's Capital University


vacuum exit
scanning magnet
Therac 20
20 MeV electrons
monitor chamber
applicator
jaws
electrons blue
photons yellow
patient
Carleton UNIVERSITY
Canada's Capital University

Flow of the process

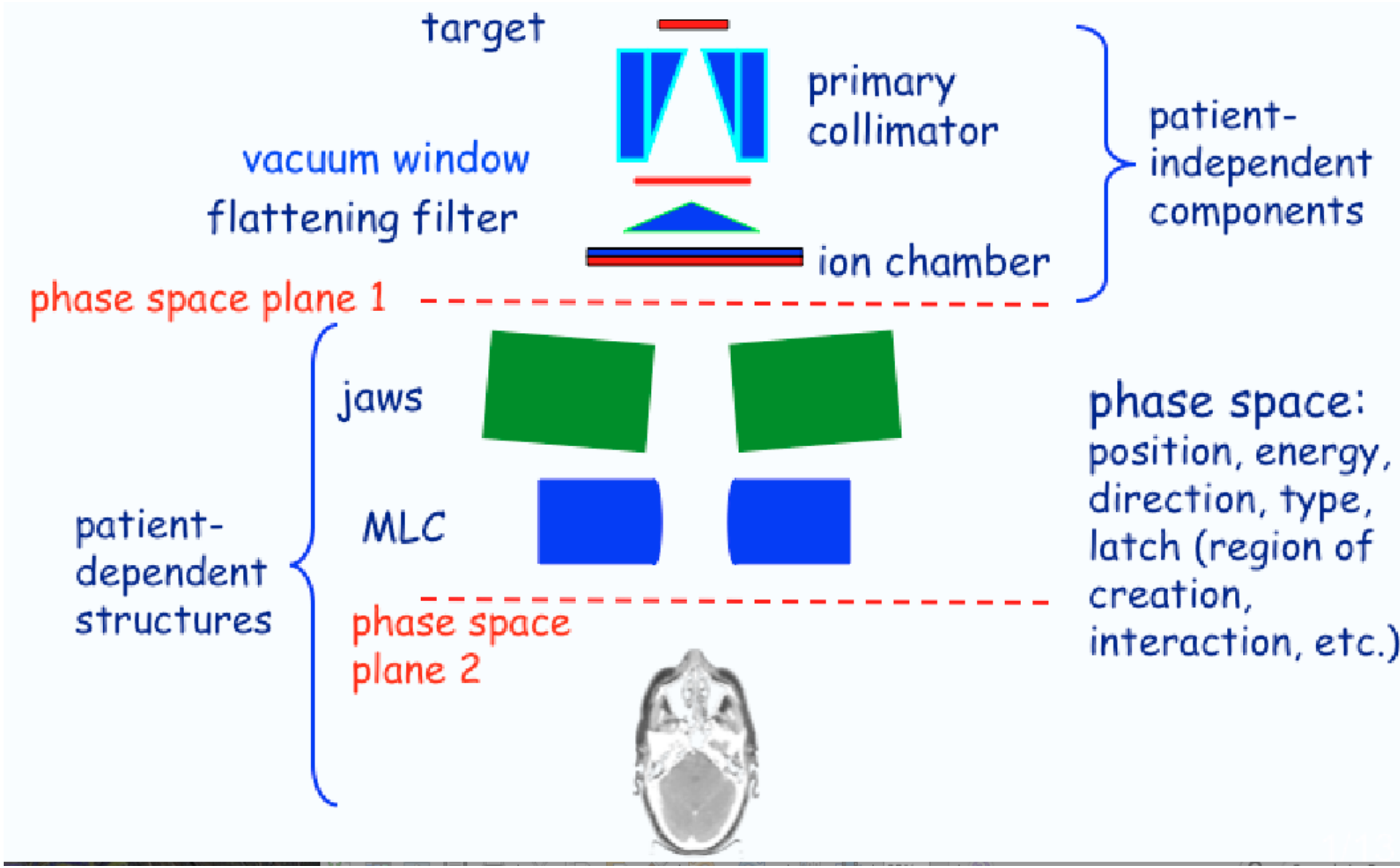


specify: defines a class of accelerators as an ordered list of CMs

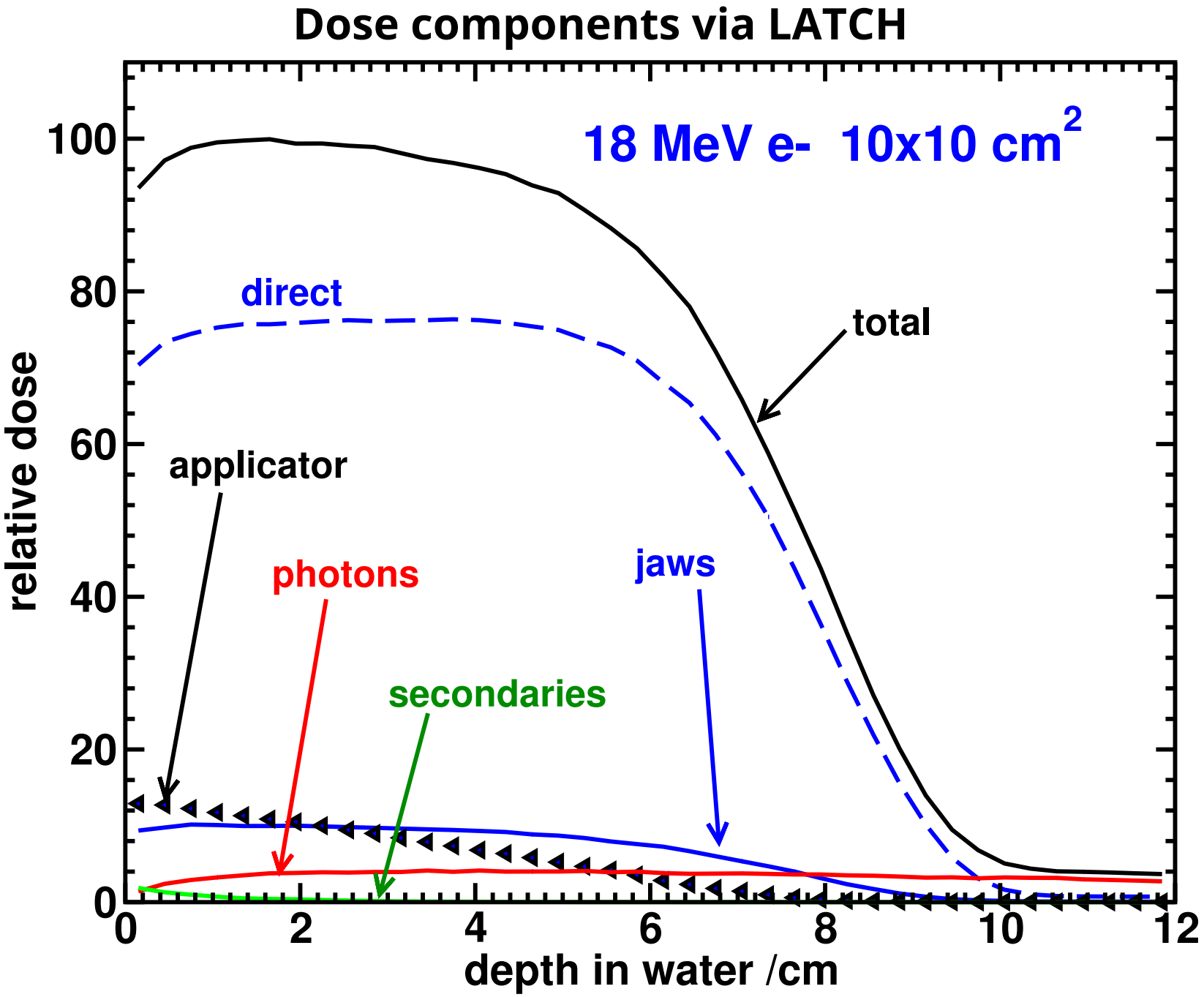
build: pulls together source code and compiles it

input file: defines specifics of one accelerator and parameters for this run

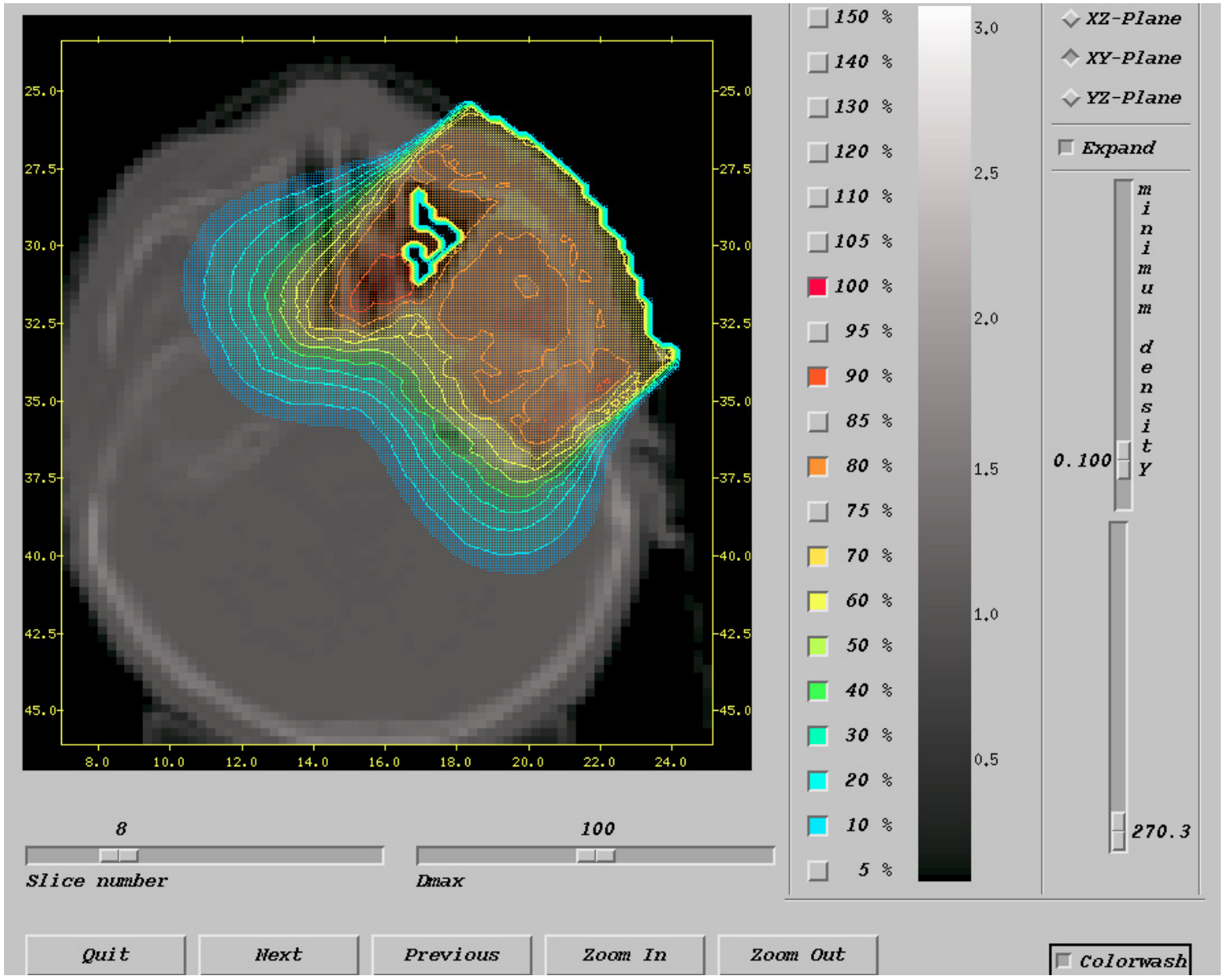
Overview of the entire process



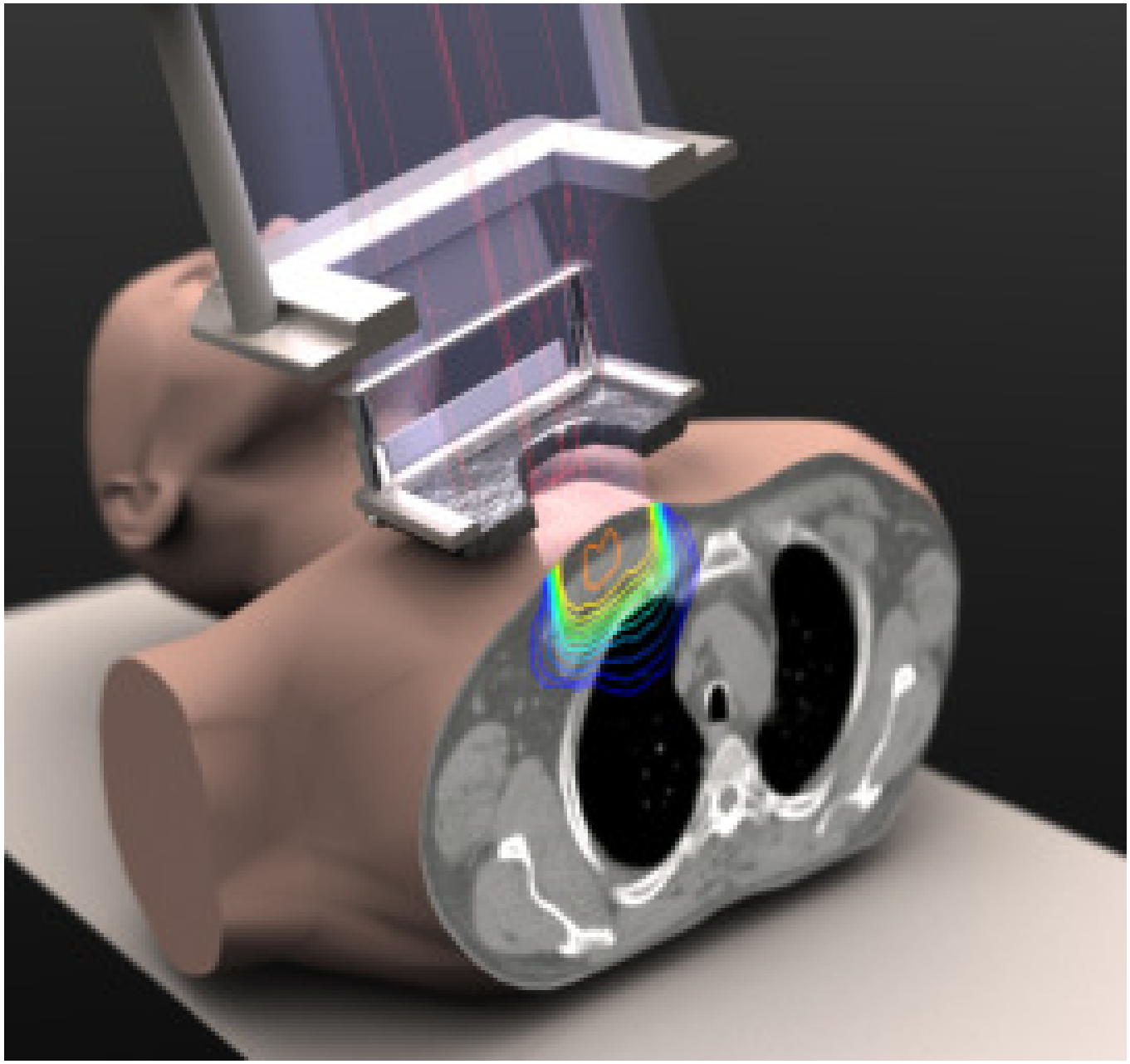
From TG105: Med Phys 34 (2007) 4818-4853.

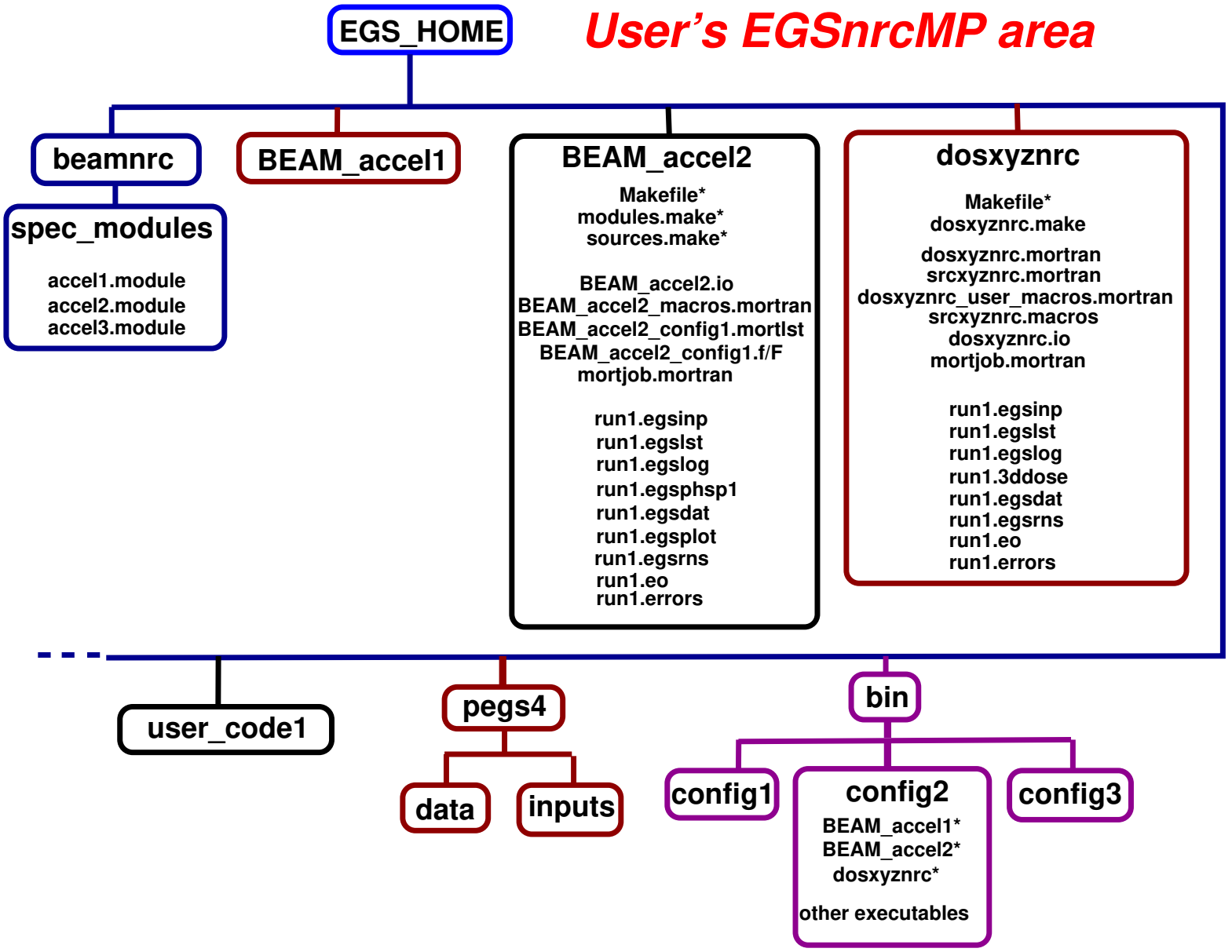

Dose components via LATCH
18 MeV e- 10x10 cm2
direct
total
applicator
photons
jaws
secondaries
relative dose
depth in water /cm
0
20
40
60
80
100
0
2
4
6
8
10
12

CT Treatment Planning: dosxyz_show



Electron beam treatment: VMC++





Parts of user's home area used with BEAMnrc. A $*$ indicates file is linked. Here, user has 2 accelerator models & 3 configs. Details of directory contents are only shown for accel2, DOSXYZnrc and config2 areas. config is "course" for the course.

mortjob.mortran

- beam_build gathers up all the needed source code and creates a file called sources.make which contains a list of all the files joined together to create one large source file called mortjob.mortran
- mortjob.mortran is then processed by mortran3 and then compiled.
- To introduce other source into a BEAM code, then modify sources.make, recalling it is overwritten if you execute beam_build.

```
# sources.make                        Generated by beam_build
#Adjust the list below to change source files used to build this accelerator

SOURCES = $(EGS_SOURCEDIR)egsnrc.macros \           $(BEAM_CODE)_cm.mortran \
          $(EGS_UTILS)timing.macros \               $(EGS_SOURCEDIR)egs_utilities.mortran \
          $(MACHINE_MACROS) $(RANDOM).macros \      $(EGS_SOURCEDIR)get_inputs.mortran \
          $(EGS_SOURCEDIR)transportp.macros \       $(EGS_SOURCEDIR)get_media_inputs.mortran \
          $(EGS_SOURCEDIR)pegs4_macros.mortran \    $(RANDOM).mortran \
          $(BEAM_HOME)beamnrc_user_macros.mortran \ $(EGS_UTILS)nrcaux.mortran \
          $(EGS_UTILS)phsp_macros.mortran $(IAEA_PHSP_MACROS)\  $(MACHINE_MORTRAN) \
          $(BEAM_CODE)_macros.mortran \             $(EGS_SOURCEDIR)egs_parallel.mortran \
          $(BEAM_HOME)beam_main.mortran\            $(EGS_SOURCEDIR)pegs4_routines.mortran \
          $(BEAM_HOME)beamnrc.mortran \             $(EGS_SOURCEDIR)egsnrc.mortran
          $(EGS_UTILS)xvgrplot.mortran \
```

Variables such as EGS_SOURCEDIR are defined in $HEN_HOUSE/specs/all_common.spec included in Makefiles via the config file. BEAM_code_macros(and cm).mortran are created by beam_build.

Tracking a Particle's History using LATCH

See section 8 of BEAMnrc manual (page 113).

The LATCH variable, associated with each particle in a simulation, is a 32-bit variable used to track the particle's history. In the input files there is an opportunity to define a mapping from geometric regions to bits using the IREGION_to_BIT variable. Thus, e.g., it is possible that bit 5 corresponds to geometric region 3, and more importantly, one bit, say 3, can correspond to multiple geometric regions, e.g., 1,5,8. Thus, although the JAWS may consist of 6 different geometric regions, they can all be associated with a single bit. Each bit is designated as follows:

bit 0 Set to 1 if a bremsstrahlung or positron annihilation event occurs in the history; 0 otherwise(not used for LATCH_OPTION = 1).

bit 1-23 Used to record the region where a particle has been and/or interacted (Note that the bit set for a region is determined by IREGION_TO_BIT for that region)

bit 24-28 Stores the region number in which a secondary particle is created; if these bits are all 0, the particle is a primary particle (not for LATCH_OPTION = 1).

cont...

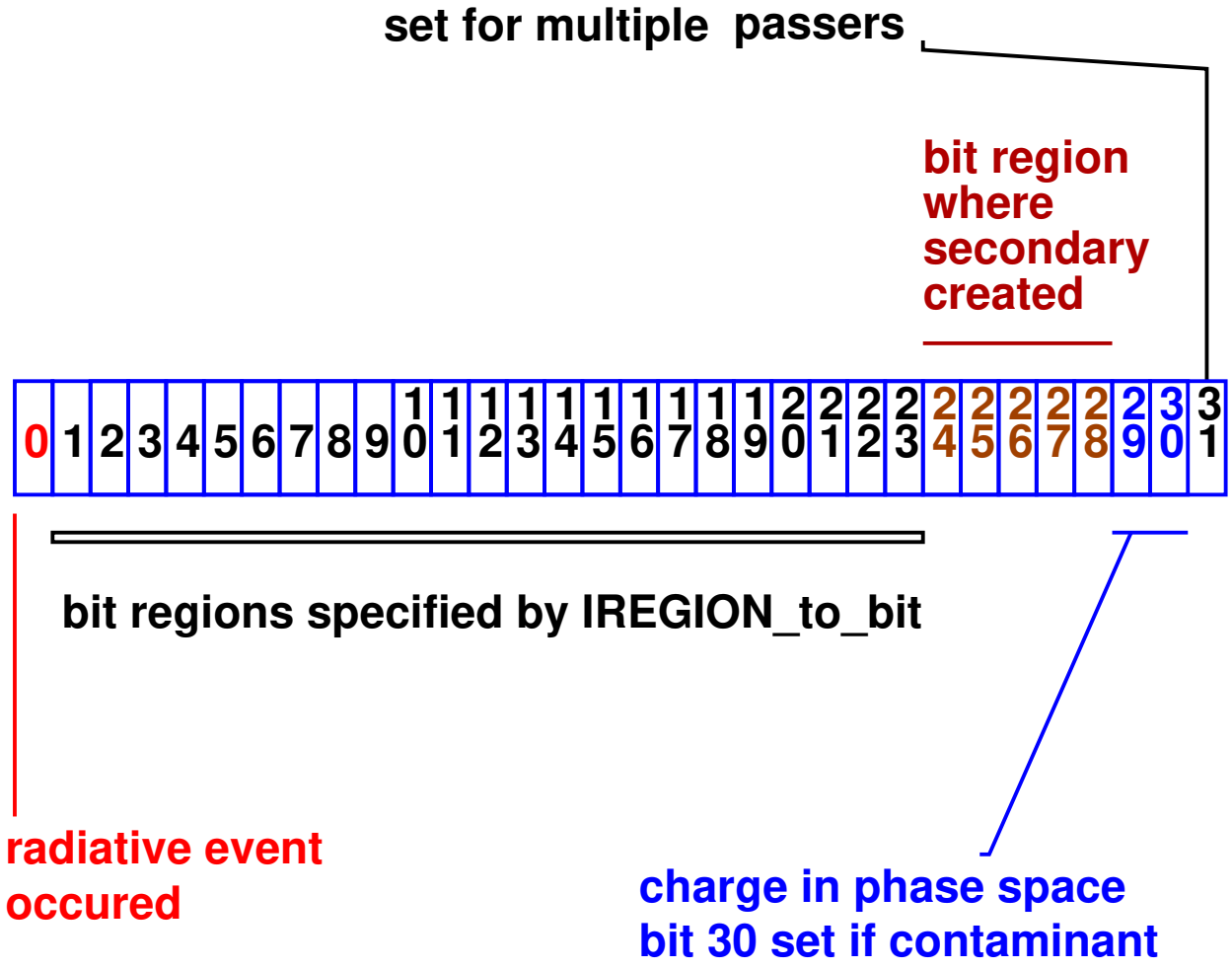
Tracking a Particle's History using LATCH (cont)

bit 29-30 Store the charge of a particle when LATCH is output to a phase space file (see BEAMnrc manual's section on phase space files). During a simulation, bit 30 is used to identify a contaminant particle but this information is not output to the phase space file. Set to 1 if the particle is a contaminant particle; 0 otherwise.
Note that if LATCH is not inherited (i.e., when LATCH_OPTION = 1), bit 30 loses its meaning.

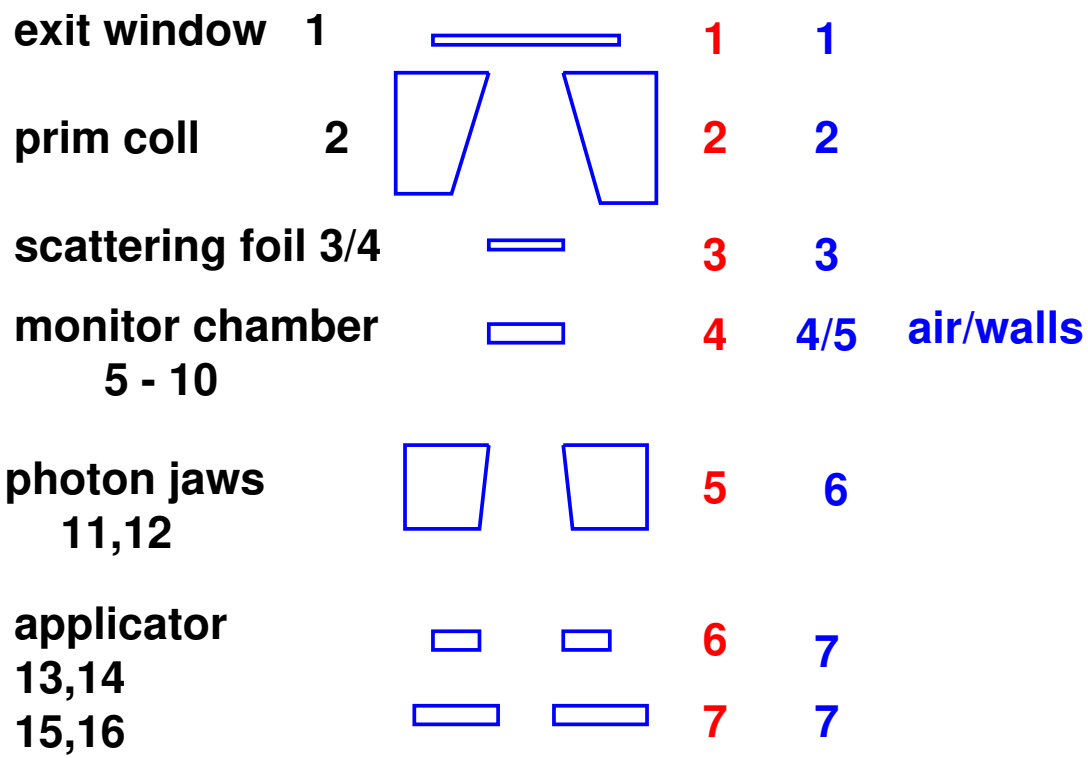
bit 31 Set to 1 if a particle has crossed a scoring plane more than once when LATCH is output to a phase space file (see BEAMnrc manual's section on phase space files)

For secondary particles, recording the region number in which they were created in bits 24-28 is equivalent to multiplying the region number by 2^{24}, or 16777216. Thus, to retrieve the region of origin of a secondary particles, the LATCH value of the particle must be divided by 16777216 (i.e., taking the value INT(LATCH/16777216)).

LATCH bit definitions
BEAM code system



Geometric vs bit regions vs dose regions



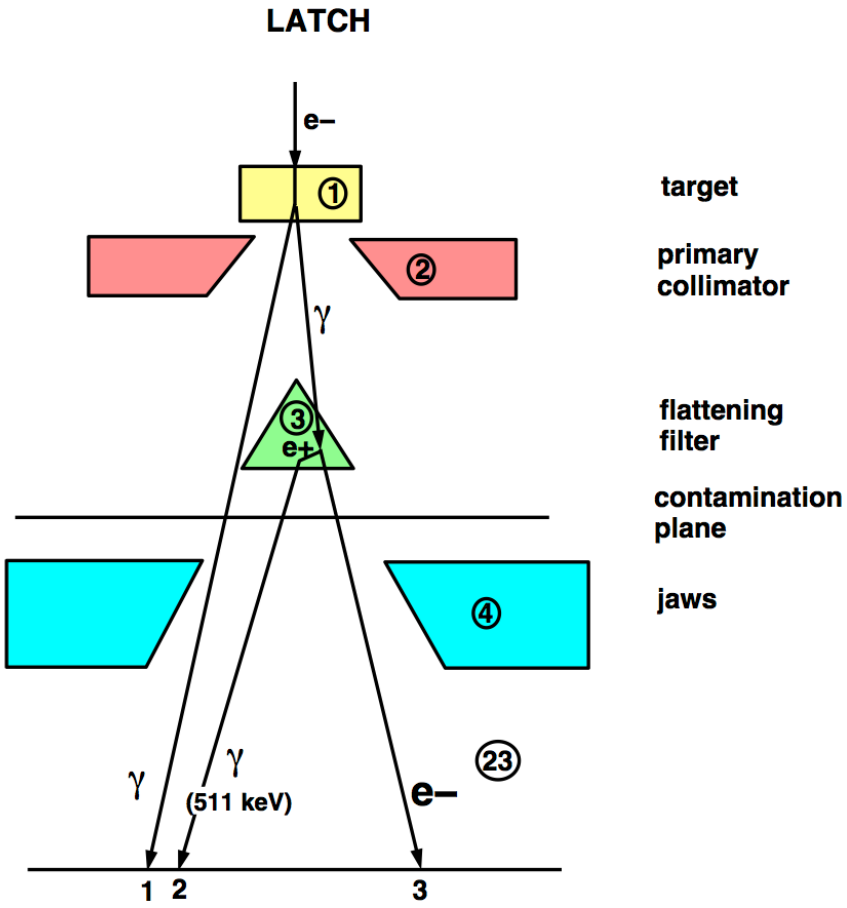
LATCH OPTION

User controls the protocol for setting LATCH using the LATCH_OPTION input variable.

LATCH_OPTION = 1 (Non-Inherited LATCH Setting): secondaries do not inherit LATCH values from primaries that created them; bits 1-23 of a secondary particle carry no information about regions its primary parent(s) has(ve) been in. Option must NOT be used if ICM_CONTAM is non-zero since that option needs bit 30. Bit 0 not set.

LATCH_OPTION = 2 (Comprehensive LATCH Setting): LATCH values passed to secondary particles from primaries that create them; bits 1-23 for a secondary particle include all regions secondary particle has been in plus those in which its primary parent(s) has(ve) been before the secondary was created; uses bits 24-28 to record where secondary particles were created and bit 0 to record whether or not a bremsstrahlung photon was involved in a particle's history.

LATCH_OPTION = 3 (Comprehensive LATCH Setting 2): similar to 2, but bits 1-23 record regions in which the particles have interacted, rather than regions they have been in. After a Compton, pair or photo-electric event, the charged particles, and in the latter case, also the fluorescent photons, have the bits 1–23 set for the region in which they are created to treat the case in which they are created below cutoff in a manner similar to being created above cutoff (where the bits would be set on first step).



Latch option	Particle	Bit 0	1	2	3	4	23	24-28	30
1 non-inherited	1	0	1	0	0	0	1	0	0
	2	0	0	0	1	0	1	0	0
	3	0	0	0	1	0	1	0	0
2 comprehensive	1	1	1	0	0	0	1	"1"	0
inherited	2	1	1	0	1	0	1	"3"	0
where been	3	1	1	0	1	0	1	"3"	1
3 comprehensive	1	1	1	0	0	0	0	"1"	0
inherited	2	1	1	0	1	0	0	"3"	0
where interacted	3	1	1	0	1	0	1	"3"	1

Simple photon accelerator model showing 3 particles reaching the phase space file after 2 electrons are incident. Contamination is defined as charged particles crossing the contamination plane. Table shows LATCH bit settings at scoring plane. Bit 0: radiative event; Bit 24-28:where created

See sec 8,p112 in BEAMnrc manual L8.

Doing it with BEAMnrc
an interactive demonstration

- use EX10MeVe accelerator model (comes with the code)

- run beamnrc_gui
  - show a compilation
  - look at inputs for accelerator
    * in-line help buttons, LATCH dose components
    * previews, how to run a job

- look at `.egsinp` file, `.egslog` and `.egslst` file

- beamdp_gui used to show spectrum (e⁻,photons), scatter plot

- demonstrate the use of `xmgrace`