Teaching the Classics of Simulation to Beginners – Panel Contribution

W. David Kelton, University of Cincinnati

Simulation Education Track
2003 Winter Simulation Conference
New Orleans

Many thanks to Ingolf Ståhl for presenting this
A Short History of Me

• Been trying to teach simulation for > 20 years
  – Universities – students from engineering, business, CS, math, stat, forestry, medicine, economics, Spanish literature ... and some strange places too
  – Non-academic – corporate seminars, industry conferences, military training
  – Experience profile from none to PhDs/MDs
  – Age/level profile including the usual 20something college students, 14-year-old high school pupils, their retiring teacher, Marine colonels
How Things Have Changed

• Really, they haven’t changed much (except software and hardware)
  – Motivations, interests of students are largely the same – want to know how to apply it, use it in a project, get a job
  – Underlying topics are pretty static – modeling, analysis, some underpinnings/theory

• Am I unique in noticing this?
• Is this good or bad?
My Intent

• Debate against myself over whether the classics should be taught
  – Define the classics
  – Dump them
  – No, they’re essential
  – The Answer
  – Different audiences, different approaches

• I agree with my fellow panelists’ points (I was late so got to read their papers before writing mine), so I won’t repeat them
What are the Classics?

• In the eye (age) of the beholder
• For me: a general-purpose procedural programming language that is not a simulation language at all
  – C, C++, any dialect of FORTRAN, maybe Java, Pascal (if it still exists)
  – Spreadsheets don’t count
  – Auxiliary subprograms for list processing, variate generation, stat accumulation, *might* count (if you wrote them yourself)
• Should we be teaching these classics?
Dump the Classics

• Nobody needs to know this stuff to build good models, use them well, do good projects
• (Almost) anyone can learn high-level, icon-based simulation software (Ray Hill) quickly
  – I agree with Ray that this is both good and bad
• Icon-based simulation software is immediately applicable to real problems
  – Students usually find and do complex class projects, some of which actually have impact
Dump the Classics (cont’d.)

• Icon-based simulation software continues to get better (not perfect), cheaper (not enough)
  – Graduates might well see it on the job
• In increasingly popular compressed OR-survey courses, you can get someplace in three weeks with icon-based software
  – But there’s no hope with the classics
• Students like high-level software
  – So they sign up for classes, which then survive
• The classics are a quill-pen waste of time
The Classics are Essential

• I actually worry about future generations’ ignorance of underlying simulation logic if we dump the classics
  – Who will write future simulation software?
• I don’t know anything about cars (and I don’t want to) but my man Tater at Tater’s Kenridge Auto Repair in Blue Ash, Ohio does, and I’m glad of it
  – We need to ensure at least some Taters for simulation-software design in the future
The Classics are Essential (cont’d.)

• Even icon-based simulators need to know something of low-level logic
  – Customize models, find errors
• Students should at least know it’s there
  – Catherine Harmonosky has a good idea with forcing students to hand in Arena .mod and .exp files, not just the graphics-based .doe model
• I always force-feed a hand simulation (event list, clock, etc.) up front, even in a high-level-software-based class (and even to MBAs …)
  – They hate it, but I know it’s Good For Them … like reading The Iliad (in Greek)
The Answer

• It depends
• I believe both sides of the argument, and am still trying to work both sides of the street
• I believe that the approach should vary with the audience, their needs, and your intended take-aways … all of which depends …
Different Audiences, Approaches

- This is all opinion only, but based on experience (both good and bad)
  - For a university class, advanced undergrad or grad, try to do a mixture (at most 20% classics)
    - Shows them that there is something underneath, and shows them something of it
    - One difficulty – programming near-illiteracy
  - For CS (and maybe some engineering students) only, spend more time on classics
    - But also do a high-level language in depth
  - For modules in OR survey classes or industry/military audience, do almost no classics
Epilogue – High-School Teaching

• Taught a one-semester course at my daughters’ high school *(pro bono)*
  – Math teacher sat in
  – My daughters wanted nothing to do with it

• OK, so they weren’t typical high-school kids
  – Went on to CalTech, Harvard, Case, Pomona, on free rides (but one is an English major ...)

• Except for ages, not much different from any of my other teaching to any type of audience
  – Though their projects tended to be sillier
  – I still did a little of The Classics
Conclusions (such as they are ...)

• Proportion of class devoted to The Classics is $\lambda \in (0, 1)$
  – Note that the interval is open on both ends – don’t spend no or all the time on them
• Can slide $\lambda$ according to audience, your intent in the class
• Seems to work from high school to PhDs/MDs to Marine colonels