UNO Software Update

Outline:

- New developments
- What we have
- What needs doing
- Where to start

New Developments

- SUNYSB/BNL proposal.
  - Contribute nodes to new communal cluster at BNL. 64 bit Opteron based. (MINOS, E949, KOPIO also interested)
  - "Interdisciplinary" - get a real CS person (student) to guide and contribute framework development.
- Water attenuation tuning of MC started.
- Focussing photo-detectors being added to MC.
- Initial point fitter developed (as a test case of the framework).

What we have: Philosophy

Balance of contradictions, descending importance

1. Long term project: getting it right takes precedence over getting it done.
2. Short term needs: need to get some studies done ASAP, but must accept sea changes in the code.
3. No reinvented wheels: prefer leveraging existing code to re-implementation
   - ROOT - external package
   - ZOOM's logging facility - incorporated package
4. Last, but not least: Minimize installation difficulties for new users

What we have: Meta

- Code management: CMT/CVS (exploring alternatives)
- Standardized code documentation: Doxygen
  http://nngroup.physics.sunysb.edu/∼pdk/
- Conventions/guidelines:
  - Coding style
  - Library version tracking
  - Memory management ←(very important)
  - Coherent system of units

What we have: Core Data Model

Sensor : PMT or other photo-detector. Knows where it is, what it is, what its neighbors are, multiple types possible, including pixelated.

Hit : collected light. A single PE, or multiple PEs collected by electronics. Knows its **Sensor**. Can backtrack from **Hit** to Reco / MC truth.

Vertex : reconstructed (or MC truth) position. Collection of out-going **Tracks**.

Track : reconstructed (or MC truth) light producing track or shower. A **Vertex**, momentum, direction, particle type, collection of potentially other **Vertex**s along the track.

Trajectory : tree of **Track** and **Vertex**, mostly useful for MC

EventModel : reconstructed (or MC truth) categorization of the entire event, contains collection of **Vertex** and **Track**. Can have multiple **EventModel**s per **Event**.

Event: recall your ZEBRA: “Event Mother Bank”

What we have: Utility

- Geometry: Where am I? How far to the closest wall, or the one I'm heading towards? Particle tracking/stepping. Based on ROOT's TGeo classes.
- Light attenuation: correction more than single exponential. TOF, timing dispersion, QE as f(wavelength).
- System of Units, Physical constants.
- Pixelation of the sphere at multiple resolutions. 2 types: Igloo and hybrid-Icosahedral, latter has lower solid angle spread after splittings.
- Minimizer interface (basis for most fitting algorithms)
- Trajectory (tree of Tracks and Vertices) traversal

What we have: MC

GEANT4 based water MC: CSIM

- Handles various detector geometries (rectangular, cylindrical)
- Input kinematics from NUANCE, G4's "gun" or trivial user code.
- Outputs **everything** including full showers and all Cherenkov photons - needs filtering code to reduce size
- Output multiple formats, but ROOT best supported

Electronics simulation: ESIM

- Post processes CSIM output
- Simple 1st PE timing model PMT
- Extensible to more elaborate photo-detectors and electronics models.

What we need

- Critical mass, almost there but at least one more serious contributer with relevant experience would make a big difference.
- Contributions needed on all levels, from least to most commitment:
  1. Come up with a question to answer, be an end user and try what exists $\Rightarrow$ direct development with your immediate requirements.
  2. Check, tune, validate: light attenuation, reconstruction efficiencies and resolutions, energy calibration, electronics.
  3. "Port" SK reconstruction and develop novel algorithms
  4. Core - participate in major design decisions, development
- Bulk MC generation and processing:
  - Soon will be in a state to doing studies with large event samples - MC for BNL VLBL: 5×10^7 seconds $= \sim 45$ days on 1.5 GHz.
  - CSU's regional software center ramping up
  - Stony Brook / BNL cluster (if we get the funding)

Where to start

1. Go to: http://nngroup.physics.sunysb.edu/∼pdk/
2. Read through the documentation
3. Ask questions: uno-software@nngroup.physics.sunysb.edu
4. Install the software
5. Ask questions: uno-software@nngroup.physics.sunysb.edu
6. Dive in, develop, debug, document
7. Ask questions: uno-software@nngroup.physics.sunysb.edu

Questions?