Building within Frameworks

Friday, March 9, 2012

IEEE/ACM Information Technology Professional Conference 2012 at TCF
The College of New Jersey
Ewing, New Jersey

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

If it makes a sound, SET IT ON SILENT
(e.g., mobiles, pagers, PDA, smartphones)

Frameworks underpin most software/hardware projects

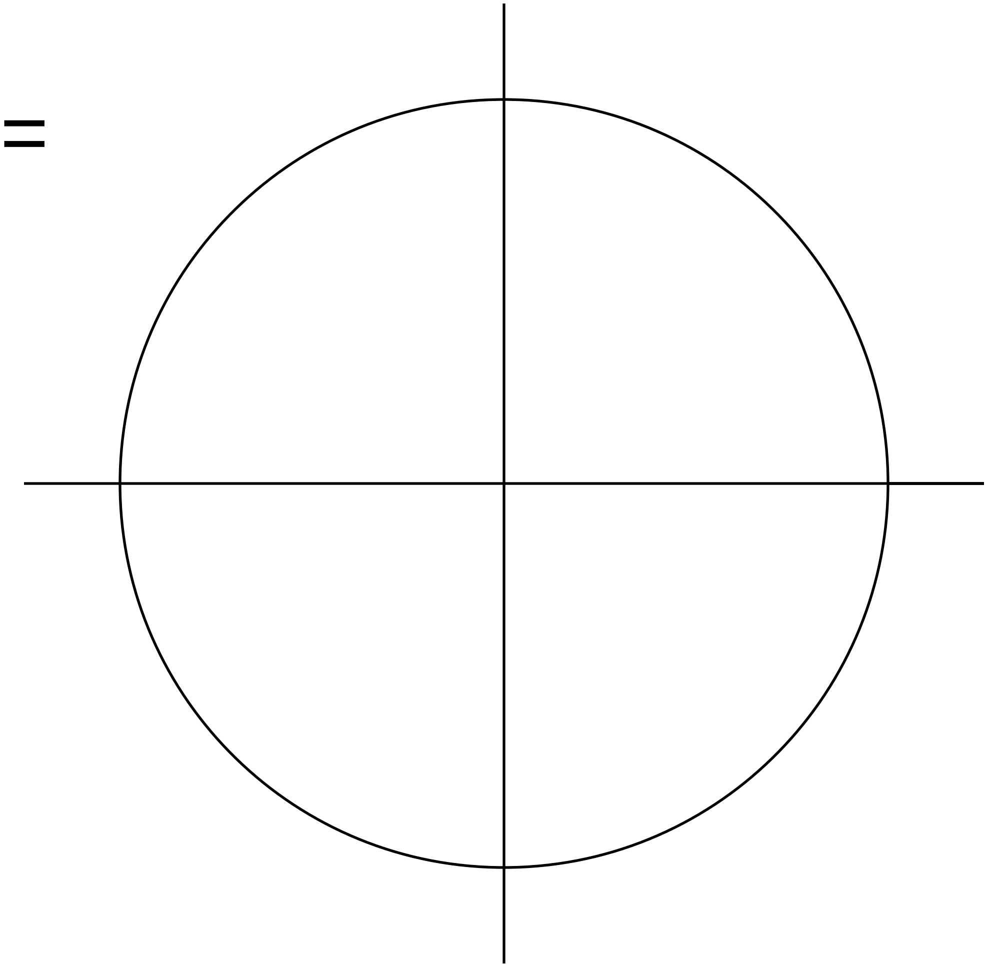
- Implementations are always a la ________
  - OS
  - CPU/GPU
  - toolkits
  - libraries
  - transaction monitors

Equivalence is not the question,
rather, the question is parsimony

- object usage
- IO
- events/threads
- interactions
- representations

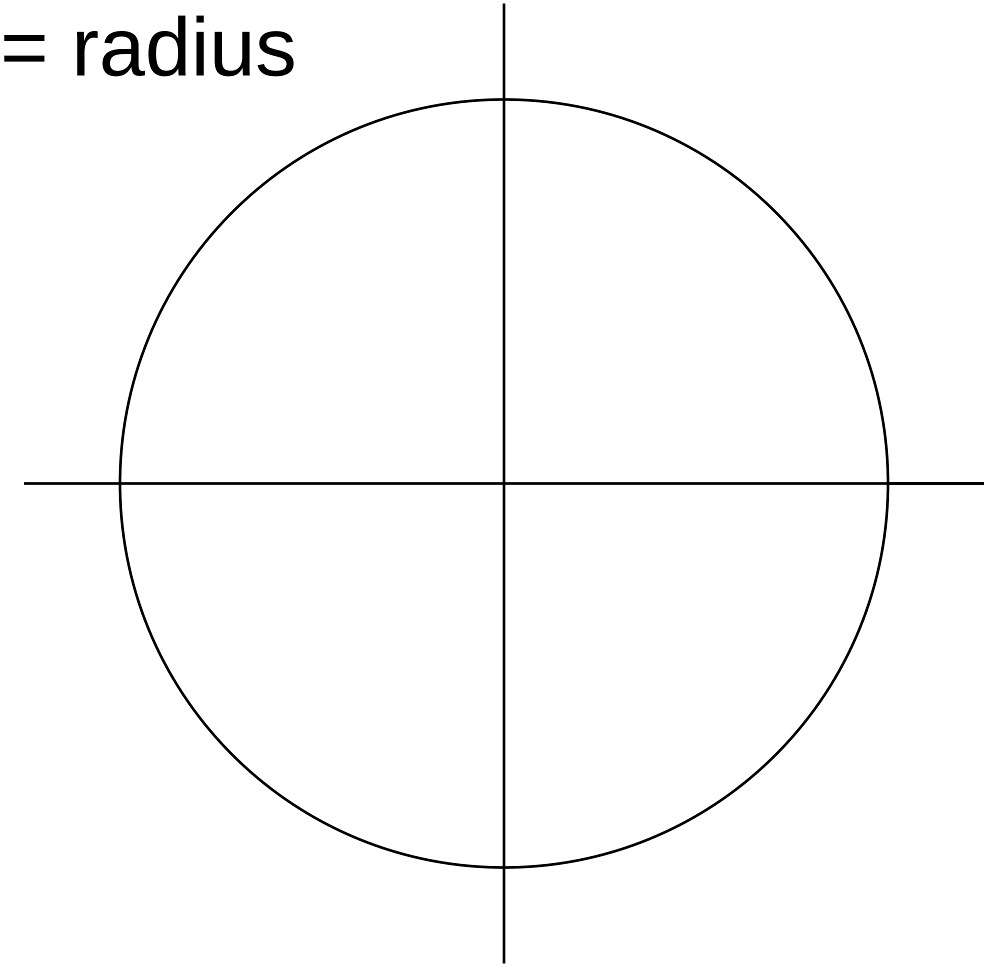
Consider coordinate systems – Cartesian/polar

- distance from origin = sqrt(x**2 + y**2)
- orthogonal axes



Coordinate Systems: Polar

- distance from origin = radius
- other calculations
  not as simple



Suitability is parsimony; not equivalence

- in either Cartesian or polar coordinates:
  - all calculations CAN be done
  - all points CAN be represented
  - difference is compactness/conciseness

Parsimony is the key to clean usage

- original FORTRAN – IBM circa 1957
  - based on scientific model - FORmula TRANslation
  - string processing - clumsy but possible

Pasimony is the key to clean usage (continued)

- COBOL – US DoD cica 1959
  - COmon Business Oriented Language
  - simple computations, string processing
  - complex calculations – clumsy

Frameworks and paradigms –

- What is intended use?
- How this is enabled.
- What is the conceptual “model”?
- Does conceptual model match intended use?

Framework choices cascade

- choices forclose alternatives
- biological example: giant panda – effectively a vegan bear with specialist adaptations
(e.g., “The Panda’s Thumb” by Gould)
- *IX: synchronous read/write API implication: threads
- OpenVMS, X-Windows, W3C File IO: asynchronous, event-based.

Events/threads Dichotomy

- synchronous easily constructed from event model
- asynchronous API requires threading
- threading is more complex that event loop

Framework extension

- should be compatible with underlying “gestalt”
- eases understanding and correct usage
- example: adding a “Parent” property to a JavaScript object

JavaScript object extension

- almost all objects can be extended
- standard definition may need extension
- example: “Parent” structure
- a mechanism to back propogate connectedness

Events/Threading

- Event-based model is more parsimonious
- Pseudo-threads (e.g., HASP)
- State-based processing (e.g., pings)
- Event based structures smaller than thread context

Lessons:

- Thoroughly understand frameworks (and implications) before adoption
- Implement usage in accord with gestalt
- Extensions should be gestalt-compatible

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Session Notes & Materials:
http://www.rlgsc.com/trentoncomputerfestival/2012/index.html