1

Summary

@ 1993 John Mylopoulos

2

What Have I Learned?

• Computation can be thought as something other than executing a sequence of statements

  • Computation as function evaluation -- functional programming

  • Computation as deductive inference -- logic programming

  • Computation as simulation and message passing -- object-oriented programming

There isn't a "universal" programming language, ...or a universal programming paradigm, for that matter!!!

@ 1993 John Mylopoulos

3

Programming As A Software Lifecycle Phase

• Programming only accounts for a small portion of the of a software system's lifecycle, in costs and/or effort (or, for that matter, importance)

For small systems, the breakdown of costs has as follows:

Specification	10%
Decomposition	20%
Coding	20%
Optimization	15%
Testing	25%
Validation	10%

For large software systems (programming-in-the-large, as opposed to programming-in-the-small) the percentage is even lower

Requirements Analysis	5%
Design	10%
Programming-in-the-small	15%
Integration	10%
Maintenance and Evolution	60%

...but programming forms the foundation of Software Engineering

@ 1993 John Mylopoulos

4

What Comes Next?

• **Information Systems Analysis and Design** -- a methodology for analysis and design of information systems; use of graphical and other notations to sketch system requirements and an initial design.

• **Programming Methodology** -- a disciplined approach to programming, whereby the programmer first constructs a specification of his program (the what) and then proceeds with an implementation (the how) by successively making the specification more procedural.

• **Software Engineering** -- a disciplined approach to software development and maintenance, software lifecycle, project management.

Software Engineering: an important topic within Computer Science right now!

@ 1993 John Mylopoulos