Alan Wassyng, Ridha Khedri,
Spencer Smith, Konstantin Kreyman

# Software Requirements Research in SQRL

Focused on Industrial Practice

## Introduction

- Brief overview of research in Software Requirements currently underway by SQRL researchers
- Industrial relevance is a core theme
- Eventual goal is to produce co-operative processes and tools - not just in requirements but throughout the software life-cycle

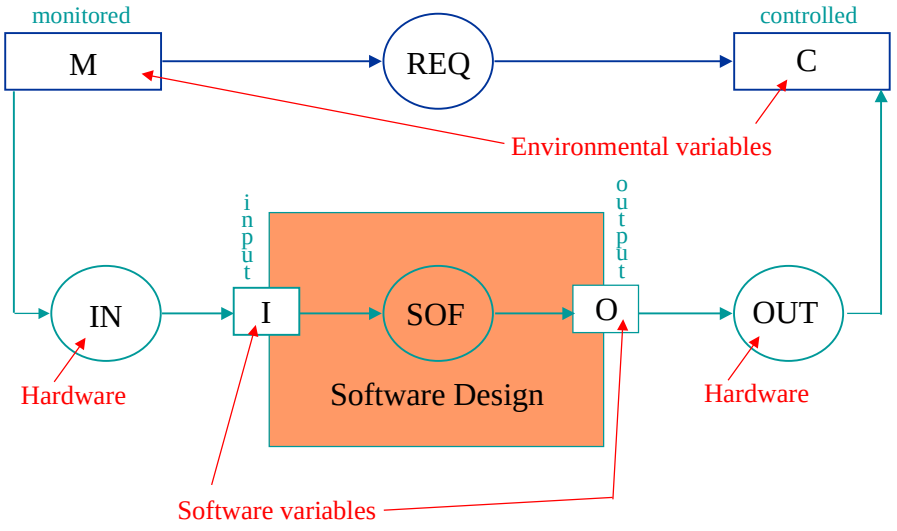
2

## Tabular Representations

- Pervasive in our representation of relations
- Thorough examination of the semantics of tables, mainly by faculty in CAS/SQRL
- A variety of software tools have been built to manipulate, analyse and convert tables
- Working on making tables more appealing to software professionals in industry

3

## Requirements Models




4

## Verification of Scenarios


- Requirements are gathered scenario by scenario (use case)
- Possible inconsistency of scenarios
- Systematically detect inconsistencies between scenarios - verification of scenarios
  - Tool (SCENATOR) based on this approach
- In order to verify scenarios we need to be able to integrate them
  - Results in major portions of the specification

5

## Decomposition of Requirements


- Various techniques for decomposing requirements documentation
  - Procedural: black-box visible; domain specific “natural language expressions”
  - Testing driven: Each “piece” of the requirements specification constitutes a system testing work assignment
    - Idea: Principal of separation of concerns (space)
    - Method: Based on relation algebra

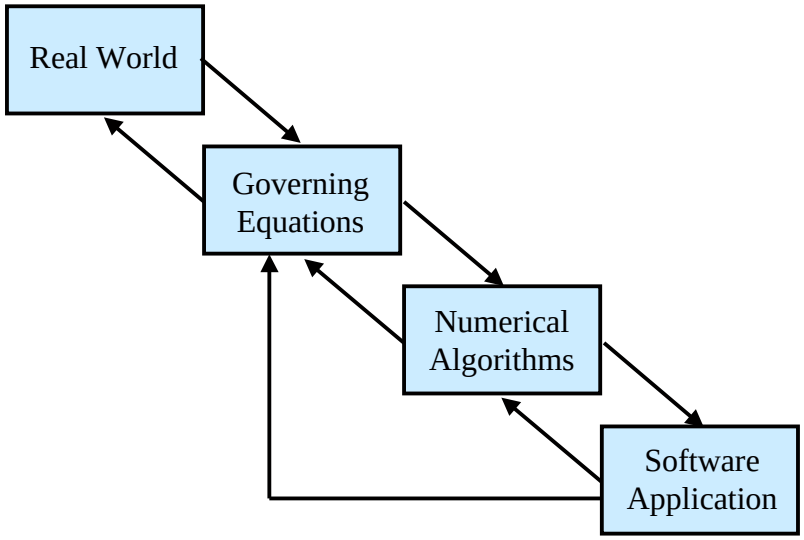
6

## Requirements for Scientific Computation


- Definition: using a computer tool to simulate real-world systems so that we can understand and predict their behaviour
- Why document requirements?
  - Communicate across different domains (math models, numerical methods, software, …)
  - Deal with complex phenomena - many details
  - Capture expert knowledge for maintainability & validation
  - Clarify assumptions & make them visible

7

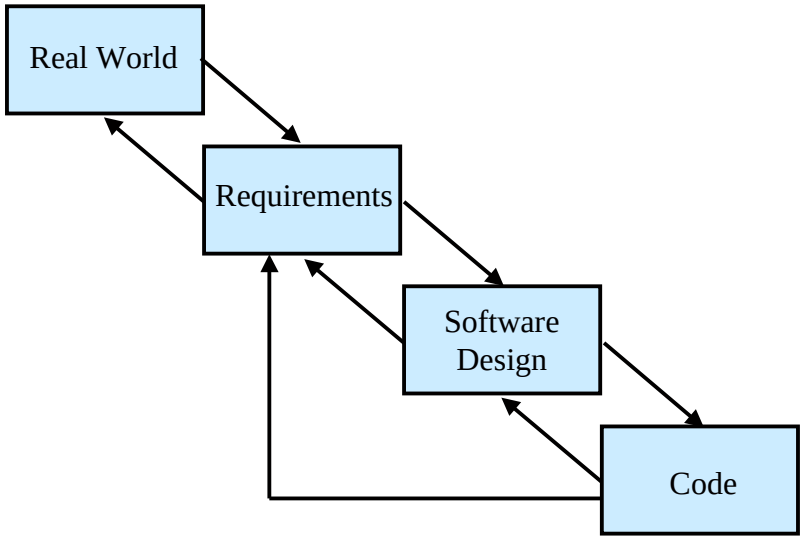
## Requirements for Scientific Computation


8

## Requirements for Scientific Computation

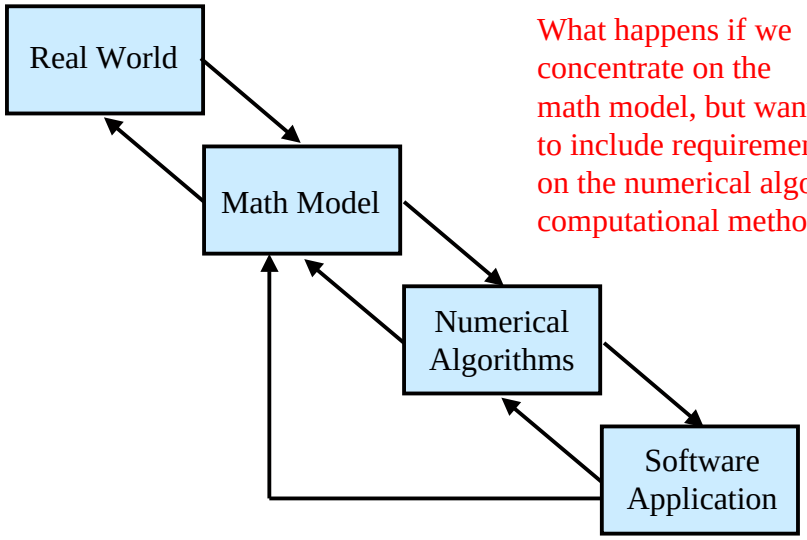
Software Quality Research Laboratory




9

## Requirements for Scientific Computation

Software Quality Research Laboratory




What happens if we concentrate on the math model, but want to include requirements on the numerical algorithms/ computational method?

10

## Documenting Math Models of Physical Systems

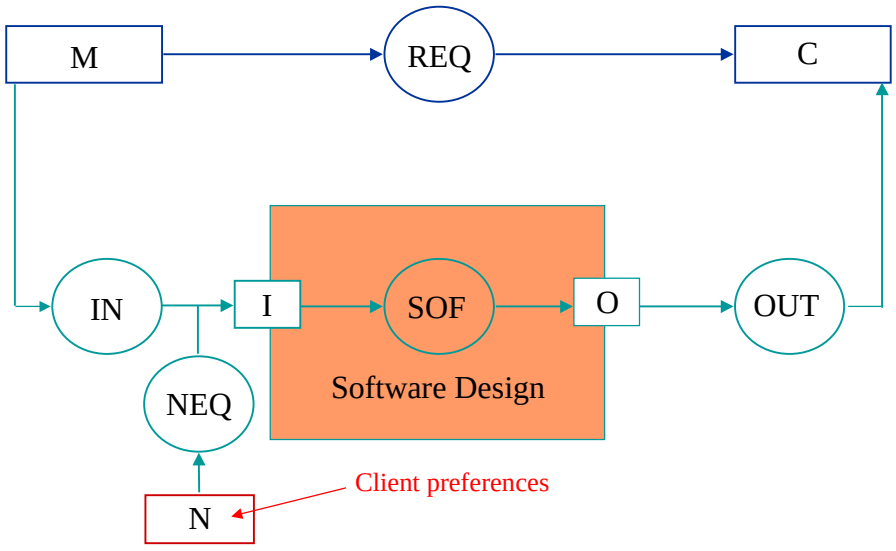
Software Quality Research Laboratory

- Include some control of the computational method
- Document quantities that represent a user's assumptions about the environment, including preferences concerning the results
- Modify the four-variable model to facilitate this

11

## Documenting Math Models of Physical Systems

Software Quality Research Laboratory




12

## Requirements for Real-Time Systems


- Research originated at OPG - safety-critical
- Two-stage requirements process
  - Mills-type black-box (history), math requirements of system, described using tables
  - FSM (discrete time, infinitely small clock-tick), math requirements of system, including system design, described using tables. Four-variable model
- Functional & performance timing requirements
- List of anticipated changes
- Self-checks
- Rationale

13

## Re-engineered Software Requirements


- Development of requirements from legacy code
- Verification that code implements re-engineered requirements

14

## Future Directions


- Math definitions of performance timing requirements
- Relationships between functional & performance timing requirements
- Executable tabular requirements
- Investigate non-functional requirements in scientific computation
- Tools

15

## Conclusion


- At least two aspects to SQRL research
  - Fundamental software engineering to improve the quality of software
  - Making our methods relevant & usable in industry
- Actively want to build/reinforce collaboration with other academics as well as industry

16