

Domain Engineering for Weather Information Services

Doreen Tuheirwe-Mukasa
dtuheirwe@cis.mak.ac.ug

Makerere University

May 2017



 WIMEA-ICT

Software Product line Engineering

- Lower software development costs, better quality, shorter time-to-market
- Focuses on reusable resources

Domain Engineering

- Build reusable assets for specific domain
- Understand domain - ability to formalize

Magnolia

- Supports domain engineering through sufficient abstraction without focus on implementation details
- Allows program language design, program transformation, optimization
- Fosters testability (parameterised tests covering data space than unit tests), verifiability of resulting code, faster implementation convergence



Rationale for Algebraic Specifications

“The practice and development of programming languages has outrun our ability to fit them into a secure mathematical framework so that they have to be described in ad hoc ways. Because these start from various points they often use conflicting and sometimes also inconsistent interpretations of the same basic term” [Strachey, 2000]



Strachey, C., 2000. Fundamental concepts in programming languages. Higher-order and symbolic computation, 13(1), pp.11-49.



Why Magnolia

- Formalized concepts (APIs) - more robust products
- Define simple, open API, add extensions as knowledge increases

Create a coherent framework for exploring the integration of algebraic specification and programming, and how this can be productively used for domain engineering

- Foundational theory for programming
- Programs are much shorter, more reconfigurable (backends for CPUs, GPUs and distributed computers)

'Magnolia supports heavy weight formalism, and enables domain concepts to be defined, represented and tested N times faster than conventional programming languages'



Structuring Concepts

Objectively:

- Brainstorming - sketch rough descriptions of the domain
- Analysis - identifies important properties within the domain
- Formulate concepts



Documenting Concepts

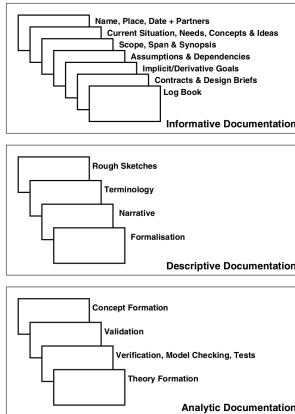


Figure: Necessary Documentation(adapted from Bjørner, 2006)



Technical Report - Possible structure

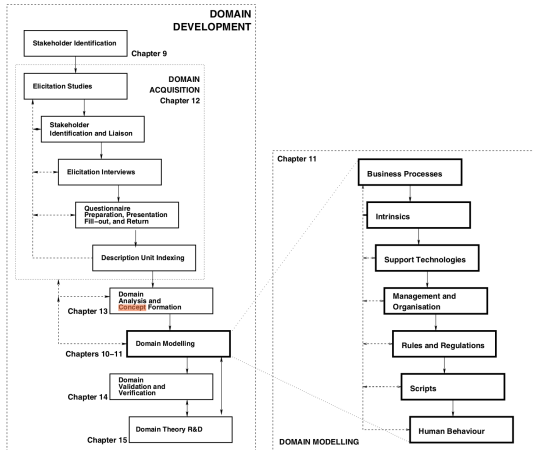


Figure: Domain Development (adapted from Bjørner, 2000)



Ongoing

- Structuring technical report
- Identifying, refining concepts



Thank you! Comments, suggestions, questions, reactions?

