NUS Seminar and Lectures
8–26 October 2018
Domain Science & Engineering

Dines Bjørner
Fredsvej 11, DK-2840 Holte, Danmark
E–Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/˜db

October 2, 2018

Contents

1 Lecture Particulars

1.1 Title: Domain Science & Engineering

- [1]: **Domain Science & Engineering Paper**[1] and **Lecture slides**[2]

1.2 Aims & Objectives

1.2.1 Aims

- The lectures are **aimed at** NUS computer scientists and 3rd–4th year MSc + PhD students

- The course **aims to** introduce participants to a new aspect of the Computing Sciences

[1] http://www.imm.dtu.dk/ dibj/2018/nus/nus2018.pdf
[2] http://www.imm.dtu.dk/ dibj/2018/nus/nus-oh.pdf

1.2.2 Objectives

The objective of the lectures is

- to inform participants about a new whole sub-discipline of the computing sciences, and
- to possibly entice them to engage in studies of some open problems of this sub-discipline.

1.3 Prerequisites

- **Motivation:** You must be interested in **software as mathematical artifacts**
- **Discrete Mathematics:** Sets, Cartesians, Algebra, ...
- **Basic Knowledge of Logic:** propositional and predicate logic
- **A Smattering of Functional Programming:**
  one of f.ex.: *Coq, Curry, Erlang, F#, Haskell, LISP, [Standard] ML, Scala, Scheme, ...*

2 Schedule

2.1 Day/Week/Month Overview – A Suggestion

```
                      OCTOBER 2018

   Monday  Tuesday  Wednesday  Thursday  Friday
   --------------------------------------------
        8    **,*9         10        11     *12
       15       16         17        18     *19
       33       23         24        25  **,*26

      ** Seminars, * Lectures 1-8   -- always: 16:00-18:00
```

2.2 Seminar and Lecture Plan

- **Week 1, Day 1, Tue.9.10, 16:00–18:00, Aud.: SR3 (COM1-02-12), Seminar 1:**
  - **Lecture 1: Introduction** [1, Sect. 1]
    - So that You know what I've been up to!
    - A prelude also to Lectures 2–5.
    - A basis for possible discussions with NUS colleagues.
  - **Lecture 2: Domain Analysis & Description, I** [1, Sect. 2]
- **Week 1, Day 2, Fri.12.10, Aud.: EC COM2-04-02, Lecture 3:**
  - **Domain Analysis & Description, II** [1, Sect. 2]
  - **Domain Facets** [1, Sect. 3.1]
- **Week 2, Day 3, Fri.19.10, 16:00–18:00, Aud.: EC COM2-04-02:**
  - **Lecture 4: From Domains to Requirements** [1, Sect. 3.2]
  - **Lecture 5: Formal Model of Prompts** [1, Sect. 3.3]
  - **Lecture 6: Axioms and Models of Mereology** [1, Sect. 3.4]

- **Week 3, Day 4, Fri.26.10, 16:00–18:00, Cerebro@COM1-0-05, Seminar 2:**
  - **Lecture 7: A Basis in Philosophy** [1, Sect. 3.5]
  - **Lecture 8: Conclusion** [1, Sects. 4–5]

2.3 Office Hours

- The lecturer, **Dines Bjørner**, is available weekday
  - mornings: 8, 10-11, 15-18, and 22-25 Oct., 9:00–12:00
  - afternoons: 9, 12, 19 and 26 Oct., 13:30–15:30

3 The Didactic Base

By a *didactic base* we shall understand the knowledge "spheres" within which we operate.

Our *didactic base* for software development is outlined below.

That base is also suggested as the *didactic base* for software engineering.

3.1 Method, Methodology and Formal Methods

- **Method:**
  - By a method we shall understand a set of **principles** for **selecting** and **applying** a number of **analysis & synthesis techniques** and **tools** in order to achieve a goal
  - where that goal here is to develop a software specification
  - whether that specification be a
    - ∗ a domain description,
    - ∗ a requirements prescription,
    - ∗ a software design and code,
    - ∗ or the first or last two, or all of these.
- **Methodology:**
  - By methodology we shall understand the study and knowledge of one or more methods.
- **Formal Methods:**
  - By a formal method we shall understand a method several of whose techniques and tools can be explained **mathematically**, such as, e.g.,
    - ∗ refinements,
    - ∗ tests, model checks, theorem proofs,
    - ∗ specification language syntax, semantics and proof systems.

The present course endows domain analysis & description with a formal method.

3.2 The Computer & Computing Sciences

3.2.1 Computer Science

- By computer science we shall understand the study and knowledge of the **properties** of the kind of phenomena that *"goes on inside" computers*.

3.2.2 Computing Science

- By computing science we shall understand the study and knowledge of how those phenomena (*"inside" computers*) can be **constructed.**

The present course is a computing science course.

3.3 The Triptych Dogma

- Before **software** can be **designed & coded**
- we must have a reasonable grasp of what is **expected & required** from that software,
- and before we can **prescribe** those **expectations & requirements**
- we must have a reasonable grasp of the **domain**, i.e., be able to **describe** it.

As a consequence we can claim that

- **Software Systems Development** can be "divided" into three phases:
  - **Domain Science & Engineering**
  - **Requirements Engineering**
  - **Software Design**

In this course we shall only consider domain analysis & description.

3.4 Informatics & IT

3.4.1 Informatics

- By **informatics** we shall understand a confluence of
  - mathematics: "pure" as well as "applied",
  - computer & computing science, and
  - software.

To us informatics is a universe of quality: correct, fit-for-purpose and pleasing

3.4.2 IT: Information Technology

- By **information technology** we shall understand a confluence of
  - hardware
  - the natural science-based technologies that *"go into making"* hardware:
    - ∗ electronics,
    - ∗ mechanics,
    - ∗ chemistry,
    - ∗ et cetera.

To us IT is a universe of quantity: faster, larger, cheaper, etc.

4 Literature

- [2, 3, Domains Analysis & Description]
- [4, 5, Domain Facets: Analysis & Description]
- [6, 7, Formal Models of Processes and Prompts]
- [8, 9, To Every Manifest Domain Mereology a CSP Expression]
- [10, 11, From Domain Descriptions to Requirements Prescriptions]
- [12, 13, Domains: Their Simulation, Monitoring and Control]
- [14, Domain Analysis & Description: Some Issues of Philosophy]
- [15, Domain Science & Engineering: A Compendium], a collection of [2, 4, 6, 8, 10, 12, 14]

Bibliography

[1] Dines Bjørner. Domain Science & Engineering – A Review of 10 Years Work and a Laudatio. In NaiJun Zhan and Cliff B. Jones, editors, *[tbd] – A Festschrift Symposium in Honour of Zhou ChaoChen*. Springer, June 2018.

[2] Dines Bjørner. A Domain Analysis & Description Method – Principles, Techniques and Modeling Languages. Research Note based on [3], Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, February 20 2018. http://www.imm.dtu.dk/ dibj/2018/tosem/Bjorner-TOSEM.pdf.

[3] Dines Bjørner. Manifest Domains: Analysis & Description. *Formal Aspects of Computing*, 29(2):175–225, Online: July 2016.

[4] Dines Bjørner. Domain Facets: Analysis & Description. May 2018. Extensive revision of [5]. http://www.imm.dtu.dk/~dibj/2016/facets/faoc-facets.pdf.

[5] Dines Bjørner. Domain Engineering. In Paul Boca and Jonathan Bowen, editors, *Formal Methods: State of the Art and New Directions*, Eds. Paul Boca and Jonathan Bowen, pages 1–42, London, UK, 2010. Springer.

[6] Dines Bjørner. Domain Analysis and Description – Formal Models of Processes and Prompts. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, 2016. Extensive revision of [7]. http://www.imm.dtu.dk/~dibj/2016/process/process-p.pdf.

[7] Dines Bjørner. Domain Analysis: Endurants – An Analysis & Description Process Model. In Shusaku Iida, José Meseguer, and Kazuhiro Ogata, editors, *Specification, Algebra, and Software: A Festschrift Symposium in Honor of Kokichi Futatsugi*. Springer, May 2014.

[8] Dines Bjørner. To Every Manifest Domain a CSP Expression — A Rôle for Mereology in Computer Science. *Journal of Logical and Algebraic Methods in Programming*, (94):91–108, January 2018.

[9] Dines Bjørner. On Mereologies in Computing Science. In *Festschrift: Reflections on the Work of C.A.R. Hoare*, History of Computing (eds. Cliff B. Jones, A.W. Roscoe and Kenneth R. Wood), pages 47–70, London, UK, 2009. Springer.

[10] Dines Bjørner. From Domain Descriptions to Requirements Prescriptions – A Different Approach to Requirements Engineering. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, 2016. Extensive revision of [11].

[11] Dines Bjørner. From Domains to Requirements. In *Montanari Festschrift*, volume 5065 of *Lecture Notes in Computer Science (eds. Pierpaolo Degano, Rocco De Nicola and José Meseguer)*, pages 1–30, Heidelberg, May 2008. Springer.

[12] Dines Bjørner. Domains: Their Simulation, Monitoring and Control – A Divertimento of Ideas and Suggestions. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, 2016. Extensive revision of [13]. http://www.imm.dtu.dk/~dibj/2016/demos/faoc-demo.pdf.

[13] Dines Bjørner. Domains: Their Simulation, Monitoring and Control – A Divertimento of Ideas and Suggestions. In *Rainbow of Computer Science, Festschrift for Hermann Maurer on the Occasion of His 70th Anniversary.*, Festschrift (eds. C. Calude, G. Rozenberg and A. Saloma), pages 167–183. Springer, Heidelberg, Germany, January 2011.

[14] Dines Bjørner. A Philosophy of Domain Science & Engineering – An Interpretation of Kai Sørlander's Philosophy. Research Note, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, Spring 2018. http://www.imm.dtu.dk/~dibj/2018/philosophy/filo.pdf.

[15] Dines Bjørner. Domain Science & Engineering – A Compendium. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, June 2018. A collection of [2, 4, 6, 8, 10, 12, 14].