Start of Lecture 1: COVER & INTRODUCTION

Lecture Notes in Software Engineering: From Domains to Requirements

Dines Bjørner
Fredsvej 11, DK-2840 Holte, Denmark
bjorner@gmail.com – www.imm.dtu.dk/~db

0. Abstract

- We present "standard" domain description and requirements prescription examples using the RAISE [RaiseMethod] Specification Language, RSL [RSL].

- The illustrated example is that of transportation networks.

  - These notes shalll serve as lecture notes for my lectures at Uppsala, Nov.8-19, 2010.

  - The present document is the ordinary "book-form"-like notes.

  - A separate document, compiled from the same files, present 11 sets of lecture slides.

  - The "funny" small numbers you see in the present document, in margins and at almost end of display lines refer to slide page numbers of the slides document.

Lecture Notes

A Tentative Lecture Schedule

1. Introduction
1.1. The Problem

- The problem to be solved by this technical note is to present in one specific formal specification language, RSL [RaiseMethod],

  − a domain description and

  − a requirements prescription developed according to the "triptych approach" [TheSEBook3].

1.2. The Triptych Approach

- The "triptych approach" calls for

  − a thorough description (cum analysis) of the domain

  − before one attempts prescribing requirements for specific software.

- As part of the triptych approach to domain engineering one starts by exploring the description ontology of specification entities:

  − *simple entities*,

  − *actions*,

  − *events* and

  − *behaviours*

- before delving into the description ontology of facets:

  − *intrinsics*,

  − *support technologies*,

  − *rules & regulations*,

  − *scripts (licenses and contracts)*,

  − *management & organisation* and

  − *human behaviour*.

• And, as part of the triptych approach to requirements engineering

  – one starts by exploring the reengineering of business processes

  – before delving into *domain requirements* concepts of

    ∗ *projection,*

    ∗ *instantiation,*

    ∗ *determination,*

    ∗ *extension* and

    ∗ *fitting* –

  followed by a number of *interface requirements* stages.

End of Lecture 1: COVER & INTRODUCTION