A Survey of Domain Engineering

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

Abstract

The practice of domain engineering as a precursor to requirements engineering in software development is gaining acceptance. In this tutorial we shall argue that before **software,** $\mathcal{S}$, can be designed, one must understand its **requirements,** $\mathcal{R}$, and that before **requirements,** can be prescribed, one must understand the application **domain,** $\mathcal{D}$. Therefore, to us, software engineering consists of three major phases: **domain engineering** in which a **domain description** is developed; **requirements engineering** in which a **requirements prescription** is developed and related to the **domain description**; and **software design** in which a **software specification** is related to the former: the $\mathcal{S}$oftware is correct with respect to the $\mathcal{R}$equirements in the context of the $\mathcal{D}$omain. In this tutorial we shall focus on domain engineering. Domain engineering, to recall, aims at producing a description of the domain: of its endurant ("data") and perdurant ("action, event and behaviour") entities. So a main activity of domain engineering is the analysis of endurant and perdurant entities. As for endurant entities we shall outline principles and techniques for describing how a domain is composed from parts, their values and types and how they relate to one another. We refer to this aspects of domain analysis as mereology. We shall, however briefly, also outline principles and techniques for describing perdurant entities: (actions, events and behaviours). The tutorial will start with the presentation of a non-trivial example: the domain description of a class of transport systems (road nets, rail nets, airline nets, shipping nets). And the tutorial will end with a discussion of how one can develop requirements from domain descriptions; how domain descriptions relate to ontology; issues of domain science: laws of domain descriptions; possible calculi of domain discoverers; who should be researching and developing domain descriptions; etcetera.

For extended tutorial notes see **http://www2.imm.dtu.dk/˜dibj/dsae-a.pdf**.

CV

Dines Bjørner was Prof. of Computing Science at The Techn. Univ. of Denmark (DTU) 1976–2007 and is now a Prof. Emeritus at DTU. He was with IBM R&D 1962–1976. At IBM Bjørner worked with Gene Amdahl (supercomputer design), John Backus (functional programming), E.F.Codd (relational databases) and at the IBM Lab., Vienna, Austria, where he worked in the small team that conceived VDM, the Vienna Development Method, the first ISO standardised formal method. Bjørner co-founded Dansk Datamatik Center 1979–1989, led many EU R&D projects including the formal spec. of a semantics for Ada, the R&D of, and compilers for CHILL and Ada, and the RAISE, Rigorous Approach to Industrial Software Engineering. 1991–1997 Bjørner was founding and first UN Director of the Macau-based UNU-IIST, the UN University's Intl. Inst. for SW Techn. In 2006 Bjørner published a three volume book "Software Engineering" with Springer and in 2009 with QingHua Press. He has published more than 120 papers, 6 books, edited 11 other books, is a member of Academia Europaea, an ACM Fellow, an IEEE Fellow, Member of the Russian Academy of Natural Science, received the John von Neuman Medal from Hungary (1993), the Masaryk Gold Medel (1996) and an honorary doctorate (2004) from the Masaryk Univ. of Brno, The Czech Republic, etc., etc., and is a Royal Knight of the Danish Flag (1984).