

Proposal for a Full-day FM 2012* Tutorial: Towards a Theory of Domain Descriptions

Dines Bjørner, Prof. Emeritus, ACM Fellow, IEEE Fellow
DTU Informatics, Techn. Univ. of Denmark, DK-2800 Kgs. Lyngby
Fredsevej 11, DK-2840 Holte, Danmark
E-Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~dibj

April 19, 2012

Abstract

The tutorial covers some of the basic ideas of domain descriptions: components and examples of fragments of domain descriptions as well as an emerging theory of domain discoverers: mental “crutches” in the form of an emerging descriptor calculus. I shall also briefly show how major components of requirements prescriptions can be “derived” from domain descriptions – thereby questioning current practice of requirements research and engineering.

The tutorial is supported by a full set of lecture notes, also in slide format: in paper form as well as in electronic form.

- Lecture 1. A Description Ontology + Example Domains** 8:30-9:10¹ + 9:20-10:00
The ontology centers around passive and active entities: parts, respectively actions, events and behaviours. Parts are analysed into either atomic or composite parts. Atomic parts are characterised by attributes. Composite parts are characterised by attributes, sub-parts and their mereology. All have unique identification.
We exemplify some domains.
- Lecture 2. Entities: Informal + Formal Descriptions** 10:30-11:10 + 11:20-12:00
Atomic and composite parts: unique identification; indivisibility of static and dynamic attributes and mereology, sub-parts are not sub-types; action signatures; event predicates; behaviour compositions.
- Lecture 3. A Calculus of Discoverers + Mereology** 14:00-14:40 + 14:50-15:30
Domain discoverers are meta-functions; survey of discoverers: PART_SORTS, PART_TYPES, UNIQUE_ID, MEREOLOGY, ATTRIBUTES, ACTION_SIGNATURES, EVENT_SIGNATURES and BEHAVIOUR_SIGNATURES. Uses of all are exemplified.
An abstract model \mathcal{M} of disjoint, embedded and overlapping parts and sub-parts. Towards an axiom system \mathcal{A} for a mereology of parts and their attributes. Satisfaction $\mathcal{A} \models \mathcal{M}$.
- Lecture 4. From Domains to Requirements + Conclusion** 16:00-16:40 + 16:50-17:30
Derivation “operators” $\mathbb{O} : \mathbb{D} \rightarrow \mathbb{R}$: projection, instantiation, determination, and extension. Is current Requirements Engineering based on an illusion?
Conclusion and Discussion.

Scope: Domain science and engineering builds on the following paradigm: (α) “Before software can be designed one must understand its requirements.” (ω) “And before one can start prescribing the requirements one must understand the domain of these.” Together with clear, mathematically supported techniques for ‘deriving’ significant parts of requirements prescriptions from domain descriptions the ω implies a revision of software engineering from two to three phases, the new phase of *domain engineering* prefixing the current phases of *requirements engineering* and *software design*. **Goal:** This, then is the ultimate aim of this tutorial: to advocate that we abandon the “twosome” in preference for a “threesome” !

*18th International Symposium on Formal Methods, CNAM: Conservatoire National des Arts et Métiers (292 rue Saint-Martin F-75141), Paris, France, August 27–31, 2012. <http://fm2012.cnam.fr/>

¹The time schedule is suggestive; it can be simply made to fit FM 1012 requirements.