DDL: A Domain Analysis and a Domain Description Langugage

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

January 9, 2026

Abstract

We informally present a domain description language. DDL.

Contents

1 Introduction

First hints of a special, formal languages for describing domains appeared in [4, 5]. A background for "merging" of the concepts of mereology and Tony Hoare's CSP was given in [3, 11]. First "formal" presentations of the special domain analysis & description method in which the language for analyxing and describing domains were published in [14, 13] followed up by the book [15]. Further papers ensued [10, 9, 20, 17, 22, 18] – as did a great number of technical reports outlining specific domains: [6, 7, 25, 8, 12, 19, 6, 7, 21, 23].

The formal analysis & description language "derived" from the RAISE [27] specification language RSL [26] – which, in turn, "derived" from VDM [24], a first, formal method for software development – which emerged from the 1973–1974 IBM Vienna Labor PL/I project [1].

We shall refer to the formal analysis & description language of this paper by `DDL`.

`DDL` emerged over the years 2010–2025 as a result of, first as the result of number of engineering case studies [16], then as the papers and book referred to above.

It is time to rigorously outline `DDL`. A formal definition is in the waiting, i.e., to be next.

2 On What Descriptions Describe

Programming languages, like `Python`, are meant to specify computations. Domain specification languages, like `DDL`, are meant to describe domains. They are not the domains [they describe]. Domain descriptions are not meant for execution! Domain descriptions may be subject, as an inital step, to software development. From domain descriptions one may be able to "derive" requirements prescriptions – as outlined in [15, *Chapter 9*] – and from requirements prescriptions one may be able to "derive" software designs – as outlined in [2].

So, what is it, in the domains, that we can and wish to describe?

That is, how do we characterize domains?

By a *domain* we shall understand a *rationally describable* segment of a *discrete dynamics* fragment of a *human assisted* reality, i.e., of the world. It includes its *endurants*, i.e., *solid and fluid entities* of *parts* and *living species*, and *perdurants*. Endurants are those quantities of domains that we can observe (see and touch), in *space*, as "complete" entities at no matter which point in *time* – "material" entities that persist, endure. Perdurants are those quantities of domains for which only a fragment exists, in *space*, if we look at or touch them at any given snapshot in *time*.

As outlined in the referenced papers and book of domain analysis & description, the study of domains is suggested performed in the context of endurants and perdurants such as summarized in Fig. 1 on the facing page.

Phenomena are what humans observe in dmains. Some phenomena are rationally descrbable – the tntities, some are not. Entities are eiter endurants or perdurants. Endurants are either solid or fluid. Solids are wither parts or are living species. Parts are either atomic or compound. Compounds are either Cartesian Parts or are Part Sets. Perdurants emerge by transcendental deduction from solids. Some endurants have internal qualities such as unique identities, mereologies and attributes. Parts are uniquely ientifiable. Mereologies express relations beween solids and are here expressed in terms of unique [part] idetifiers. Attributes are then what "give life" to endurants[1].

We list some of the concepts mentioned above – and graphed in Fig. 1 on the next page.

- `domain`
- `entity`
- `endurant`
- `external quality`
- `solid (entity)`
- `fluid (entity)`
- `part`
- `living species`
- `atomic part`
- `compound part`
- `composite (Cartesian part)`
- `part set`
- `internal quality`
- `unique identifier`
- `mereology`
- `attributes`
- `perdurant`
- `channel`
- `action`
- `event`
- `behaviour`

[1] Examples of part attributes could be: street segments [links] have lengths and are either one or two way – and open or closed for traffic, etc.; street intersections [hubs] have states: set of pairs of link itifiers, and state spaces; automobiles have velocity, etc.; all parts have histories: time-stamped recordings of events such as automobile leaving a link entering a hub, etc.

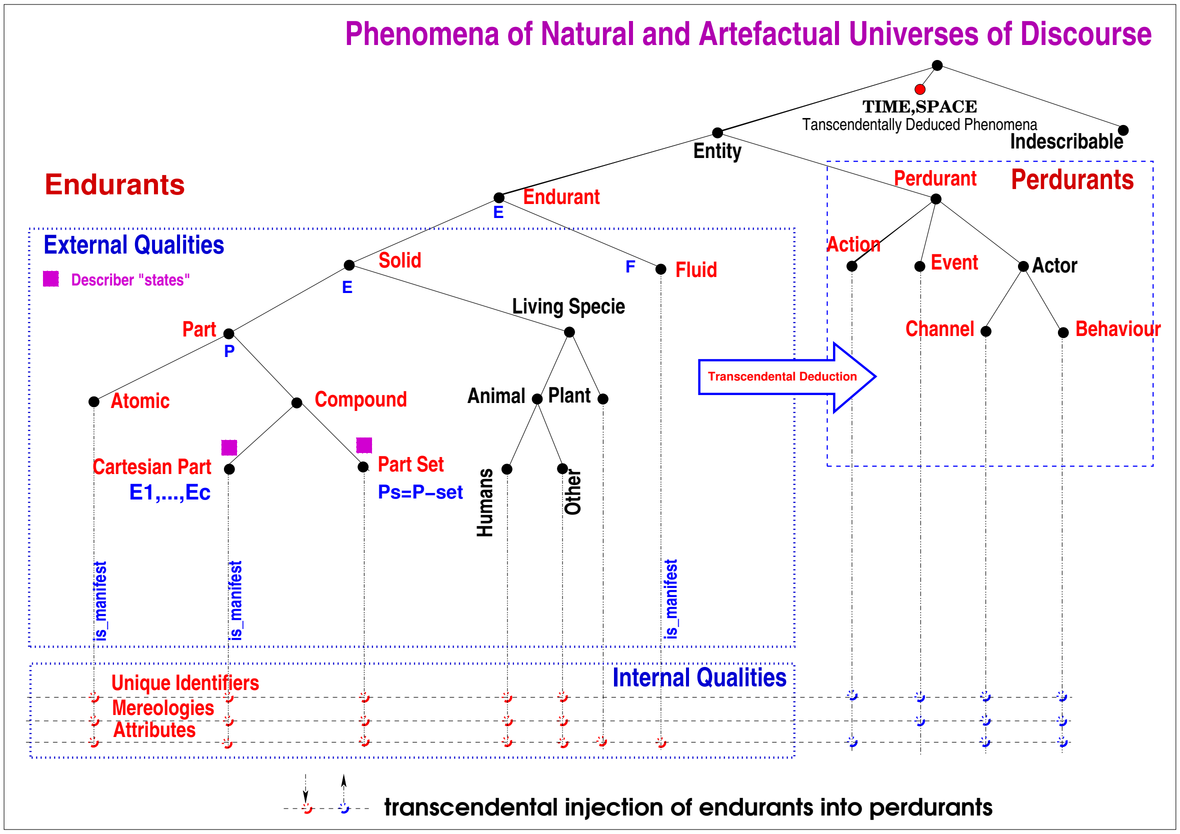

Figure 1: A Domain Analysis & Description Ontology

3 Description Units

3.1 Types

3.2 Values

3.3 Axioms

3.4 Channels

3.5 Variables

4 Expressions

5 Definitions

6 Closing

7 Bibliography

References

[1] H. Bekič, D. Bjørner, W. Henhapl, C. B. Jones, and P. Lucas. A Formal Definition of a PL/I Subset. Technical Report 25.139, IBM Laboratory, Vienna, December 1974.

[2] Dines Bjørner. *Software Engineering*, volume 1: Abstraction and Modelling, vol. 2: Specification of Systems and Languages, vol. 3: Domains, Requirements and Software Design of *Texts in Theoretical Computer Science, the EATCS Series.* Springer, 2006. Chapters 12–14 of vol.2 primarily authored by Christian Krog Madsen.

[3] 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. www.imm.dtu.dk/ dibj/bjorner-hoare75-p.pdf.

[4] Dines Bjørner. Domain Science & Engineering – *From Computer Science to The Sciences of Informatics, Part I of II: The Engineering Part*. *Kibernetika i sistemny analiz*, 2(4):100–116, May 2010.

[5] Dines Bjørner. Domain Science & Engineering – *From Computer Science to The Sciences of Informatics Part II of II: The Science Part*. *Kibernetika i sistemny analiz*, 2(3):100–120, June 2011.

[6] Dines Bjørner. Weather Information Systems: Towards a Domain Description www.imm.dtu.dk/ dibj/2016/wis/wis-p.pdf. Technical Report: Experimental Research, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, November 2016.

[7] Dines Bjørner. What are Documents ? www.imm.dtu.dk/ dibj/2017/docs/docs.pdf. Research Note, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, July 2017.

[8] Dines Bjørner. Container Terminals. www.imm.dtu.dk/ dibj/2018/yangshan/maersk-pa.pdf. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, September 2018. An incomplete draft report; currently 60+ pages.

[9] Dines Bjørner. Domain analysis & description - the implicit and explicit semantics problem www.imm.dtu.dk/ dibj/2017/bjorner-impex.pdf. In Régine Laleau, Dominique Méry, Shin Nakajima, and Elena Troubitsyna, editors, Proceedings Joint Workshop on *Handling IMPlicit and EXplicit knowledge in formal system development (IMPEX)* and *Formal and Model-Driven Techniques for Developing Trustworthy Systems (FM&MDD)*, Xi'An, China, 16th November 2017, volume 271 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–23. Open Publishing Association, 2018.

[10] Dines Bjørner. Domain Science & Engineering – A Review of 10 Years Work and a Laudatio. www.imm.dtu.dk/ dibj/2017/zcc/bjorner-zcc-80.pdf. In NaiJun Zhan and Cliff B. Jones, editors, *Symposium on Real-Time and Hybrid Systems – A Festschrift Symposium in Honour of Zhou ChaoChen*, LNCS 11180, pp. 6184. Springer Nature Switzerland AG, June 2018.

[11] Dines Bjørner. To Every Manifest Domain a CSP Expression www.imm.dtu.dk/ dibj/2016/mereo/mereo.pdf. *Journal of Logical and Algebraic Methods in Programming*, 1(94):91–108, January 2018.

[12] Dines Bjørner. *An Assembly Plant Domain – Analysis & Description*, `www.imm.dtu.dk/ dibj/2021/assembly/assembly-line.pdf`. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, September 2019.

[13] Dines Bjørner. Domain Analysis & Description – Principles, Techniques and Modeling Languages. *ACM Trans. on Software Engineering and Methodology*, 28(2):66 pages, March 2019.

[14] Dines Bjørner. Domain Analysis & Description – Principles, Techniques and Modelling Languages. www.imm.dtu.dk/ dibj/2018/tosem/Bjorner-TOSEM.pdf. *ACM Trans. on Software Engineering and Methodology*, 28(2), April 2019. 68 pages.

[15] Dines Bjørner. *Domain Science & Engineering – A Foundation for Software Development*. EATCS Monographs in Theoretical Computer Science. Springer, Heidelberg, Germany, 2021. A revised version of this book is [18].

[16] Dines Bjørner. *Rigorous Domain Descriptions. A compendium of draft domain description sketches carried out over the years 1995–2024. Chapters cover:*

- *Graphs,*
- *Rivers,*
- *Canals,*
- *Railways,*
- *Road Transport,*
- *The "7 Seas",*
- *The "Blue Skies",*
- *Credit Cards,*
- *Weather Information,*
- *Documents,*
- *Urban Planning,*
- *Swarms of Drones,*
- *Container Terminals,*
- *A Retailer Market,*
- *Double-entry Bookkeeping,*
- *Shipping,*
- *Stock Exchanges,*
- *Web Transactions, etc.*

This document is currently being edited. Own: www.imm.dtu.dk/ dibj/2021/dd/dd.pdf, Fredsvej 11, DK-2840 Holte, Denmark, November 15, 2021.

[17] Dines Bjørner. Domain Modelling. In Jonathan Bowen et al., editor, *Theories of Programming and Formal Methods: Essays Dedicated to Jifeng He on the Occasion of His 80th Birthday*, Lecture Notes in Computer Science, Festschrift. Spriger, Heidelberg, Germany, August 2023. https://www.imm.dtu.dk/ dibj/2023/FEA/hjf.pdf and https://www.imm.dtu.dk/ dibj/2023/final/HeJiFeng.pdf.

[18] Dines Bjørner. Domain Modelling – A Primer. A short and significantly revised version of [15]. xii+202 pages[2], May 2023.

[19] Dines Bjørner. Double-entry Bookkeeping. Research, Institute of Mathematics and Computer Science. Technical University of Denmark, DK-2800 Kgs.Lyngby, Denmark, August 2023. `http://www.imm.dtu.dk/ dibj/2023/doubleentry/dblentrybook.pdf`.

[20] Dines Bjørner. Pipelines: A Domain Science & Engineering Description. In *FSEN 2023: Fundamentals of Software Engineering, Teheran, Iran, May 3–5*. `www.imm.dtu.dk/ dibj/2023/tehran/tehran.pdf`, 2023.

[2] This book is currently being translated into Chinese by Dr. Yang ShaoFa, IoS/CAS (Institute of Software, Chinese Academy of Sciences), Beijing and into Russian by Dr. Mikhail Chupilko and his colleagues, ISP/RAS (Institute of Systems Programming, Russian Academy of Sciences), Moscow

[21] Dines Bjørner. Banking – A Domain Description. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, 18 August 2025. https://www.imm.dtu.dk/˜dibj/2025/banking/main.pdf.

[22] Dines Bjørner. Domain analysis & description – a tutorial. In *ICTAC 2025 Proveedings*, Lecture Notes in Computer Science. Springer, November 2025.

[23] Dines Bjørner. Transport – A Domain Description. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, 12 June 2025. https://www.imm.dtu.dk/˜dibj/2025/transport/main.pdf.

[24] Dines Bjørner and Cliff B. Jones, editors. *The Vienna Development Method: The Meta-Language*, volume 61 of *LNCS*. Springer, Heidelberg, Germany, 1978.

[25] Dines Bjørner. Urban Planning Processes. www.imm.dtu.dk/ dibj/2017/up/urban-planning.pdf. Research Note, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, July 2017.

[26] Chris W. George, Peter Haff, Klaus Havelund, Anne Elisabeth Haxthausen, Robert Milne, Claus Bendix Nielsen, Søren Prehn, and Kim Ritter Wagner. *The RAISE Specification Language*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1992.

[27] Chris W. George, Anne Elisabeth Haxthausen, Steven Hughes, Robert Milne, Søren Prehn, and Jan Storbank Pedersen. *The RAISE Development Method*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1995.