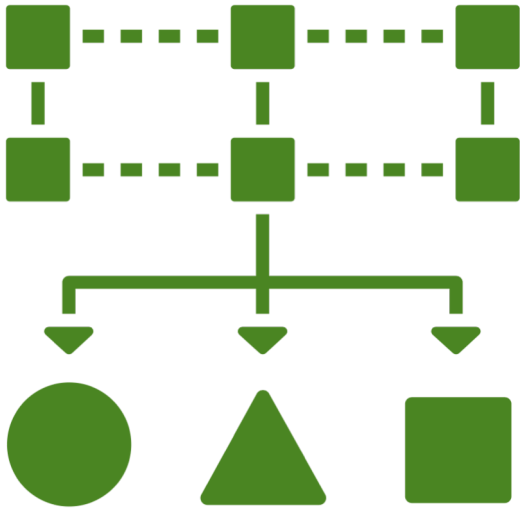
Towards a classification of DSLs



Mikhail Barash University of Turku Finland

OOPSLE 2020

Existing classifications

R. Lämmel, "Software languages"

paradigm	imperative, functional, object-oriented, logic
type system	static typing, dynamic typing, duck typing, ...
purpose	programming, querying, modeling, logging, ...
generality	
representation	strings, trees, graphs
notation	textual, markup, visual
declarativeness	rule-based, constraint-based, ...

Comparison and classification of programming languages	Babenko et al.	1975
A classification system for visual programming languages	Burnett, Baker	1994
On the problem of computer language classification	Anureev et al.	2008
Development of the computer language classification portal	Shilov et al.	2011
New developments of the computer language classification knowledge portal		2013
Taxonomic system for computer languages	https://hopl.info/keyset.html	2006
DSL classification	Langlois, Jitia, Jounenne	2007
Classification of DSLs	M. Brambilla	2010
Method and tool support for classifying SL with Wikipedia	Lämmel et al.	2013
A taxonomy of domain-specific aspect languages	Fabry et al.	2015

Taxonomic system for computer languages

Regnum	Phylum	Class	Order	Family	Genus		R P C O F G S	Regnum	Phylum	Class	Genera		
												name	parent	code
	Algorithmic	Conversational	JOSS family	Generation of Joss I			1 1 1 1 1 0 0	1000000	1100000	1110000	Generation of Joss I	1111000	1111100
				Generation of Joss II			1 1 1 1 2 0 0	1000000	1100000	1110000	Generation of Joss II	1111000	1111200
			BASIC family	Dartmouth basics			1 1 1 2 1 0 0	1000000	1100000	1110000	Dartmouth basics	1112000	1112100
				Street basics			1 1 1 2 2 0 0	1000000	1100000	1110000	Street basics	1112000	1112200
				Technical basics			1 1 1 2 3 0 0	1000000	1100000	1110000	Technical basics	1112000	1112300
				Modern Basics			1 1 1 2 4 0 0	1000000	1100000	1110000	Modern Basics	1112000	1112400
			Conversational Coeval				1 1 1 3 0 0 0	1000000	1100000	1110000	Conversational Coeval	1110000	1113000
			Fortran family	Generation of Fortran I-III	True FORTRAN I-IIIs		1 1 2 1 1 1 0	1000000	1120000	1121100	True FORTRAN I-IIIs	1121100	1121110
					Fortran Coeval		1 1 2 1 1 2 0	1000000	1120000	1121100	Fortran Coeval	1121100	1121120
				Generation of Fortran IV	FORTRAN IV standard		1 1 2 1 2 1 0	1000000	1120000	1121200	FORTRAN IV standard	1121200	1121210
					Non Standard FIV		1 1 2 1 2 2 0	1000000	1120000	1121200	Non Standard FIV	1121200	1121220
					FORTRAN 66 standard		1 1 2 1 3 1 0	1000000	1120000	1121300	FORTRAN 66 standard	1121300	1121310
				Generation of Fortran 77	FORTRAN 77 standard		1 1 2 1 4 1 0	1000000	1120000	1121400	FORTRAN 77 standard	1121400	1121410
					Non Standard F77		1 1 2 1 4 2 0	1000000	1120000	1121400	Non Standard F77	1121400	1121420
				Generation of FORTRAN 90/95	FORTRAN 90/95 standard		1 1 2 1 5 1 0	1000000	1120000	1121500	FORTRAN 90/95 standard	1121500	1121510
					F		1 1 2 1 5 2 0	1000000	1120000	1121500	F	1121500	1121520
					Non Standard F90/95		1 1 2 1 5 3 0	1000000	1120000	1121500	Non Standard F90/95	1121500	1121530
					HPF		1 1 2 1 5 4 0	1000000	1120000	1121500	HPF	1121500	1121540
			Algol family	Generation of Algol 58/IAL	True Algol58s		1 1 2 2 1 1 0	1000000	1120000	1122100	True Algol58s	1122100	1122110
					IAL Coeval	Jovials	1 1 2 2 1 2 1	1000000	1120000	1122120	Jovials	1122120	1122121
						Other IAL Coeval	1 1 2 2 1 2 2	1000000	1120000	1122120	Other IAL Coeval	1122120	1122122
				Generation of Algol 60	True ALGOL60s		1 1 2 2 2 1 0	1000000	1120000	1122200	True ALGOL60s	1122200	1122210
					CPL Algols	CPLs, BCPLs and Bs	1 1 2 2 2 2 1	1000000	1120000	1122200	CPLs, BCPLs and Bs	1122200	1122221
						Cs	1 1 2 2 2 2 3	1000000	1120000	1122200	Cs	1122200	1122223
						OO Cs	1 1 2 2 2 2 4	1000000	1120000	1122200	OO Cs	1122200	1122224
					Wirth Algols	Algol Ws	1 1 2 2 2 3 1	1000000	1120000	1122200	Algol Ws	1122230	1122231
						Pascals	1 1 2 2 2 3 2	1000000	1120000	1122200	Pascals	1122230	1122232
						Modulas	1 1 2 2 2 3 3	1000000	1120000	1122200	Modulas	1122230	1122233
						Oberons	1 1 2 2 2 3 4	1000000	1120000	1122200	Oberons	1122230	1122234
						Adas	1 1 2 2 2 3 5	1000000	1120000	1122200	Adas	1122230	1122235
					Other Algol 60s		1 1 2 2 2 4 0	1000000	1120000	1122200	Other Algol 60s	1122200	1122240
				Generation of Algol 68	True ALGOL68s		1 1 2 2 3 1 0	1000000	1120000	1122300	True ALGOL68s	1122300	1122310
					Partial A68 only		1 1 2 2 3 2 0	1000000	1120000	1122300	Partial A68 only	1122300	1122320
			PL/I Languages	IBM PL/Is			1 1 2 3 1 0 0	1000000	1120000	1123000	IBM PL/Is	1123000	1123100
				XPLs			1 1 2 3 2 0 0	1000000	1120000	1123000	XPLs	1123000	1123200
				Multics PL/Is			1 1 2 3 3 0 0	1000000	1120000	1123000	Multics PL/Is	1123000	1123300
				SIMPLs			1 1 2 3 4 0 0	1000000	1120000	1123000	SIMPLs	1123000	1123400
				Other			1 1 2 3 5 0 0	1000000	1120000	1123000	Other	1123000	1123500
						Early Autocodes	1 1 2 4 1 1 1	1000000	1120000	1124110	Early Autocodes	1124110	1124111

Taxonomic system for computer languages https://hopl.info/keyset.html 2006

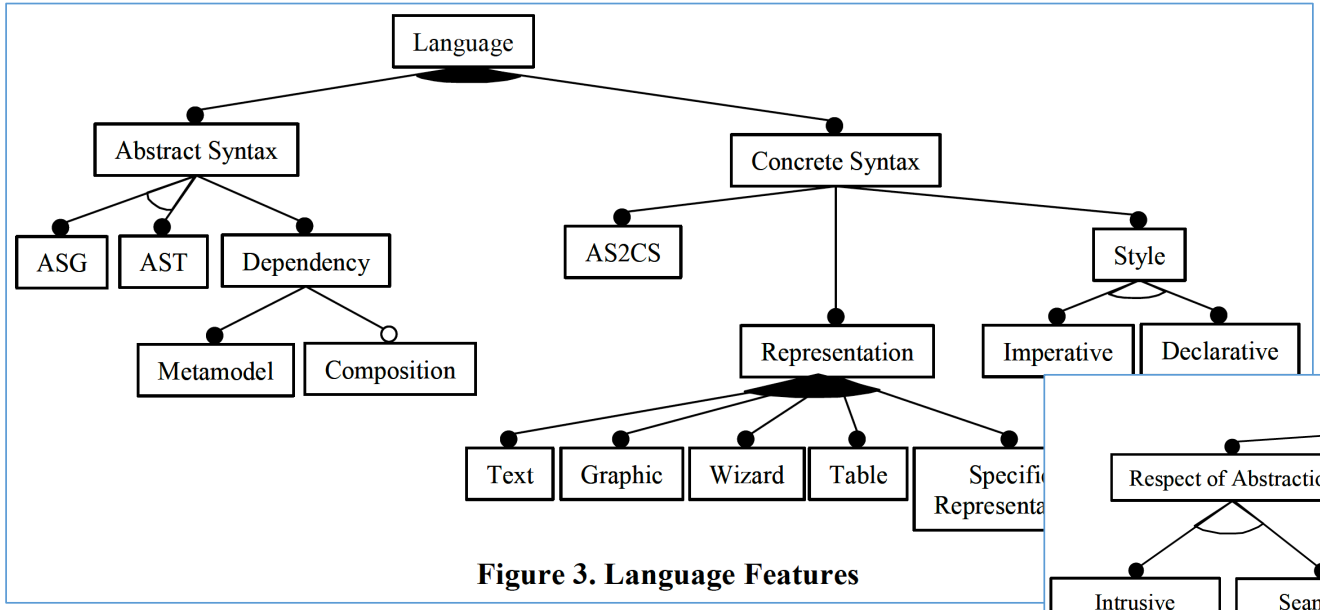


Figure 3. Language Features

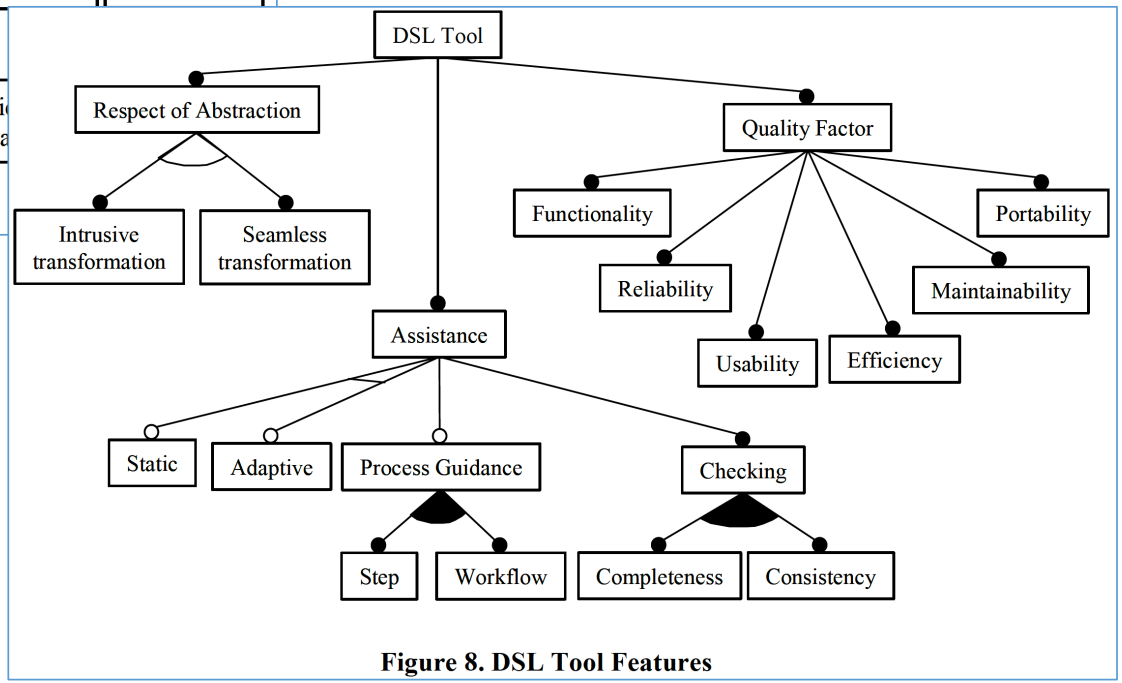


Figure 8. DSL Tool Features

DSL classification **Langlois, Jitia, Jounenne** **2007**

TABLE I. DSLs FOR MACHINE LEARNING IN BIG DATA

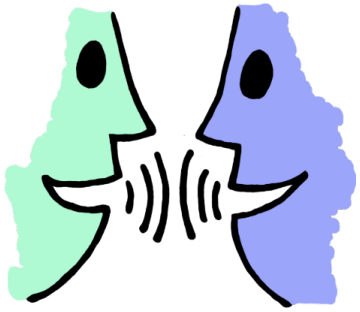
Language Name	*Requirements / Programming / Modeling*	*Textual / Graphical*	*Internal / External*	*Dynamically typed / Statically typed*	*Imperative / Declarative*	*Translation / Interpretation*	*Target Platform / Execution Engine*	*Descriptive model / Prescriptive model*	*Supports Vector (V) / Matrix (M) / Graph (G) operations*	*Supports Parallel operations*	*Supports Distributed (D) / Cloud (C) computing*
OptiML [62]	Programming	Textual	Internal (Scala)	Statically typed	Declarative	Translation	-	-	V/M/G	Yes	-/-
ScalOps [71]	Programming	Textual	Internal (Scala)	Statically typed	Declarative	Translation	-	-	V/M/G	Yes	D/C
Pig Latin [51]	Programming	Textual	External	Dynamically typed	Imperative	Translation	Pig Latin compiler / Apache Pig	-	V/M/-	Yes	D/C
SCOPE [13]	Programming	Textual	External	Dynamically typed	Declarative	Translation	SCOPE Compiler / Cosmos Execution Environment	-	V/M/-	Yes	D/C
Sawzall [52]	Programming	Textual	External	Statically typed	Imperative	Interpretation	Sawzall compiler / Sawzall engine (proprietary)	-	V/M/-	Yes	D/C
VisuML [6]	Modeling	Graphical	External	-	-	-	-	Descriptive	-	-	-
Graphical models [27]	Modeling	Graphical	External	-	-	-	-	Descriptive	-	-	-

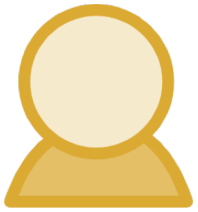
TABLE II. FRAMEWORKS FOR MACHINE LEARNING IN BIG DATA

Framework name	*Textual / Graphical*	*Languages*	*Supports Vector (V) / Matrix (M) / Graph (G) operations*	*Supports Parallel operations*	*Supports Distributed (D) / Cloud (C) computing*
Infer.net [47]	Textual	.NET framework languages	V/M/-	Yes	-/-
Graphlab [41]	Textual	C++, Python	V/M/G	Yes	D/C
TensorFlow [1]	Textual	C++, Python	V/M/G	Yes	D/C

A survey on DSLs for machine learning in big data **Portugal et al.** **2016**

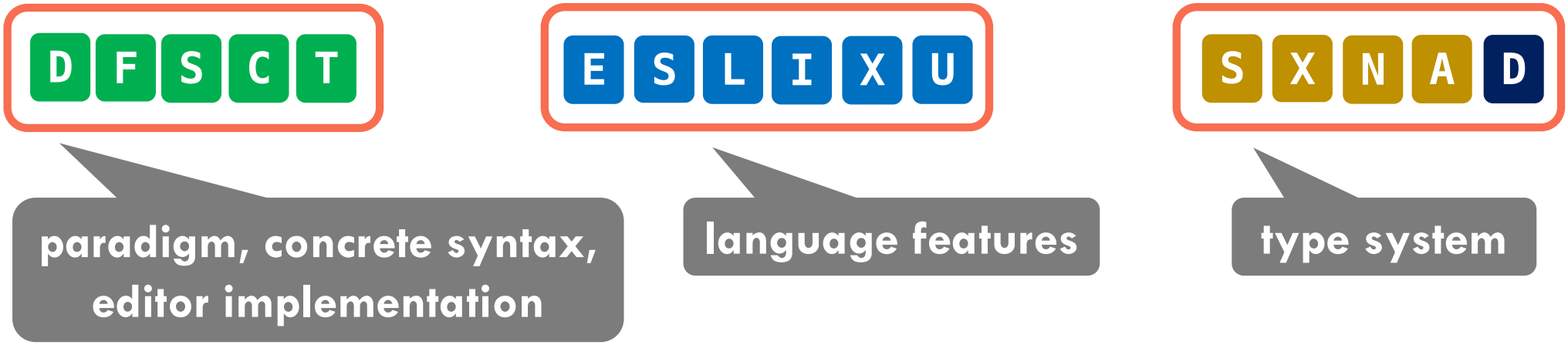
language engineer





language engineer

A mnemonic classification



DFSCT:ESLIXU:SXNAD

A mnemonic classification

Paradigm, concrete syntax

- **D** declarative
- **I** imperative
- **F** has functional features
- **E** allows side effects
- **S** “structured” syntax
- **U** “unstructured” syntax
- **H** homoiconicity
- **C** C-style blocks
- **P** Pascal-style blocks
- **Y** Python-style blocks
- **T** textual syntax
- **N** non-textual syntax
- **J** editor is projectional
- **M** editor supports math formulae
- **G** editor supports graphical notation
- **X** editor supports tabular notation
- **A** non-English-based syntax

A mnemonic classification

- **E** arithmetic expressions
- **S** subroutines
- **L** repetition statements
- **I** conditionals
- **X** exception handling
- **U** user-defined data types
- **B** objects, entities
- **G** generics

Language features

- **M** modularity
- **R** reflection
- **P** contracts
- **C** concurrency, parallelism
- **T** type system

A mnemonic classification

- **S** strong
- **W** weak
- **X** explicit
- **M** implicit
- **N** nominal
- **P** property-based
- **A** static
- **Y** dynamic

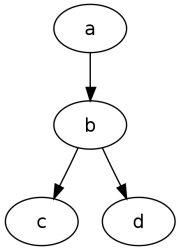
- **D** dependent types
- **L** linear types
- **I** intersection types
- **U** union types
- **E** existential types
- **G** gradual typing

Z **Z** **Z** unspecified

Examples

DOT language / Graphviz DSCT

```
digraph graphname {
        a -> b -> c;
        b -> d;
}
```



UML DNG:Z

XML DSHPT:M

XSLT DFSHPT:MESLI

Markdown DUT

JSON DSHCT

YAML DSHYT

Language composition

M. Voelter et al., “DSL Engineering”

>>	language extension
@@	language embedding
++	language reuse
&&	language referencing

XML	DSHPT:M	XSLT	DFSHPT:MESLI
XML	DSHPT:M	XSLT	DSHPT:M>>F:ESLI
CSS	DCT	SASS	DCT>>S:ESLIBM

Decoding the encoding

Isomorphism?

Equivalence classes?

DFSPT:MESLI

declarative | has functional features | "structured" syntax

Pascal-style blocks | textual syntax | modularity | conditionals

arithmetic expressions | subroutines | repetition statements

```
namespace A
sub X
      block
            if 2+2 > 4 then
                  loop 10
                        block
                        end block
                  end loop
            end if
      end block
end sub
```

```
<xsl:function name="X" xmlns:ns="A">
  <block1>
    <xsl:if test="2+2>4">
      <xsl:for-each select="…">
        <block2>
        </block2>
      </xsl:for-each>
    </xsl:if>
  </block1>
</xsl:function>
```