Robotics DSL Zoo

An Effort to Structure, Consolidate and Harmonize DSL Developments in Robotics

Arne Nordmann Nico Hochgeschwender Sebastian Wrede

October 20st 2014

CORE Lab
Research Institute
for Cognition and Robotics

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Universität Bielefeld

Motivation

Goal: Online resource to *"structure, consolidate and harmonize domain-specific language developments in robotics."*

Targeted to:

1. **DSL Users:**
   - domain experts, looking for method and tool support
   - provide means to assess availability and usability of DSLs
2. **DSL Developers:**
   - robotics system developers and integrators
   - provide an overview on state of the art, common solutions and best practices
   - foster scientific exchange and community building inside the domain

Outline

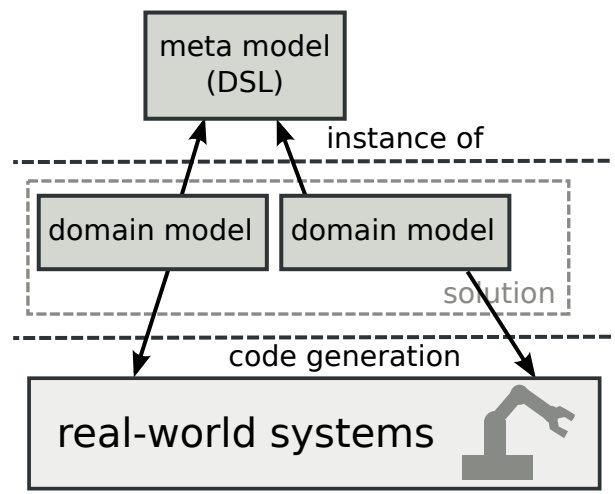
Prerequisites

Domain-Specific Languages

"programming language or **executable specification language** *that offers, through* **appropriate notations and abstractions**, *expressive power focused on, and usually restricted to, a* **particular problem domain** *. . . abstractions* **natural/suitable for the stakeholders** *who specify that particular concern."* [1]

```
Robot Fancy {
RobotBase FancyBase {
    inertia_params{...}
    children { link1 via jA}
}
link link1 {
    id = 1
    inertia_params {
        mass = 1.0
        CoM = (0.5, .0, .0)
        Ix=0.0025  Iy=0.084 Iz=0.084
           Ixy=0.0  Ixz=0.0 Iyz=0.0
```

[2]



Survey Process

1. **Scanned 6 robotics conferences** for the keywords *"domain-specific language"*, *"domain-specific modeling language"*, *"generative programming"*, *"specification language"*, *"description language"*, and *"code generation"*.
2. **Scanned 2 software conferences** for the keywords *"robot"* and *"robotics"*.
3. $\Rightarrow$ Raw list of **210** unique publications

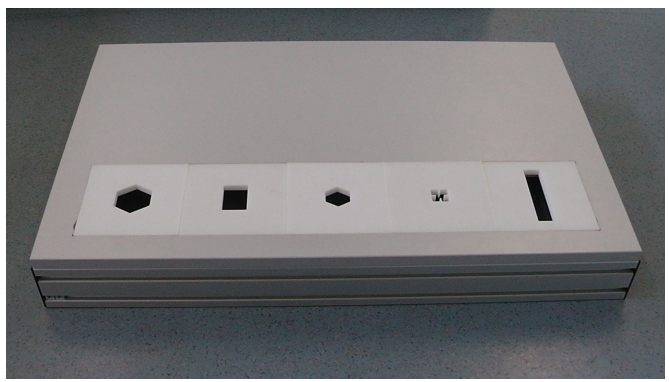
Survey Process

1. **Scanned 6 robotics conferences** for the keywords *"domain-specific language"*, *"domain-specific modeling language"*, *"generative programming"*, *"specification language"*, *"description language"*, and *"code generation"*.
2. **Scanned 2 software conferences** for the keywords *"robot"* and *"robotics"*.
3. $\Rightarrow$ Raw list of **210** unique publications

Domain Analysis and Example

Problem: Limit the scope of the survey

1. Precision Placement Test (PPT) from RoboCup@Work
   1. Robot Structures
   2. Coordinate Representation and Transformation
   3. Perception
   4. Reasoning and Planning
   5. Manipulation and Grasping
   6. Coordination
   7. Motion Control
   8. Architecture
   9. (Software) Components



Filtering

Filtered 210 publications:

1. Targets relevant concern of robotics
2. Technical aspects
   1. must provide a language definition or meta-model
   2. must be textual (internal or external) or graphical languages
   3. must provide an example of their concrete syntax (notation)
   4. should explain how a mapping to a target technology is achieved

$\Rightarrow$ **41** publications left after filtering

Filtering

Filtered 210 publications:

1. Targets relevant concern of robotics
2. Technical aspects
   1. must provide a language definition or meta-model
   2. must be textual (internal or external) or graphical languages
   3. must provide an example of their concrete syntax (notation)
   4. should explain how a mapping to a target technology is achieved

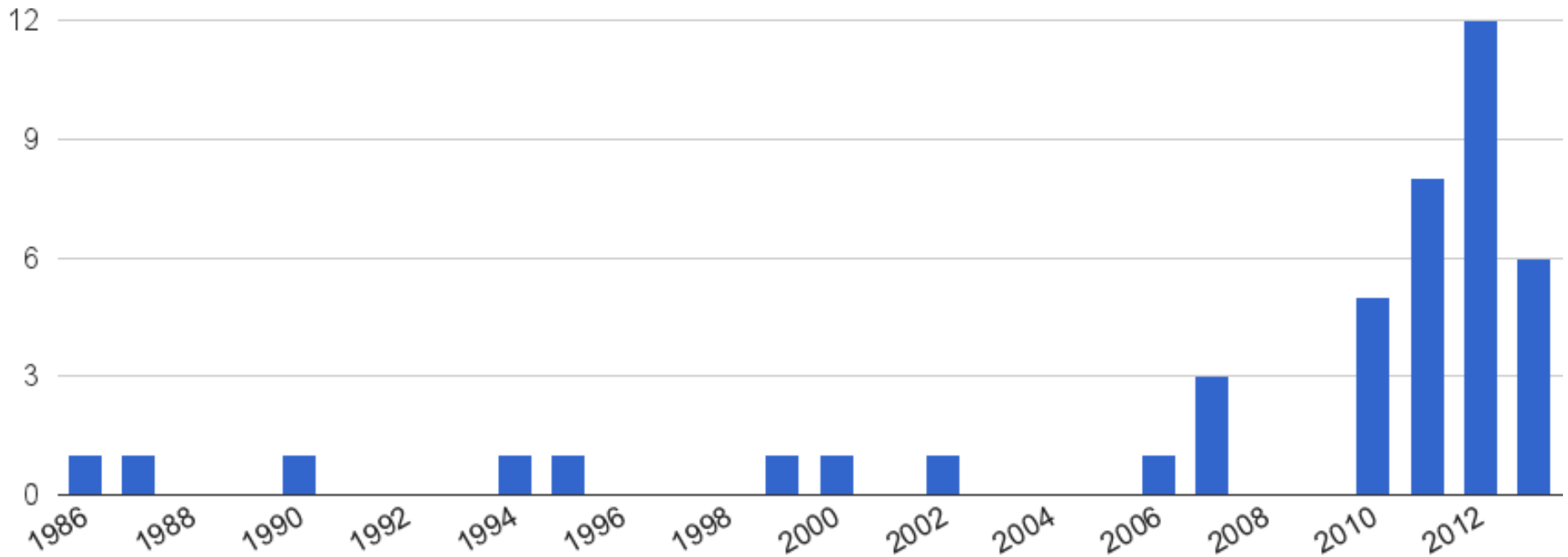
$\Rightarrow$ **41** publications left after filtering

Survey

Some facts and figures

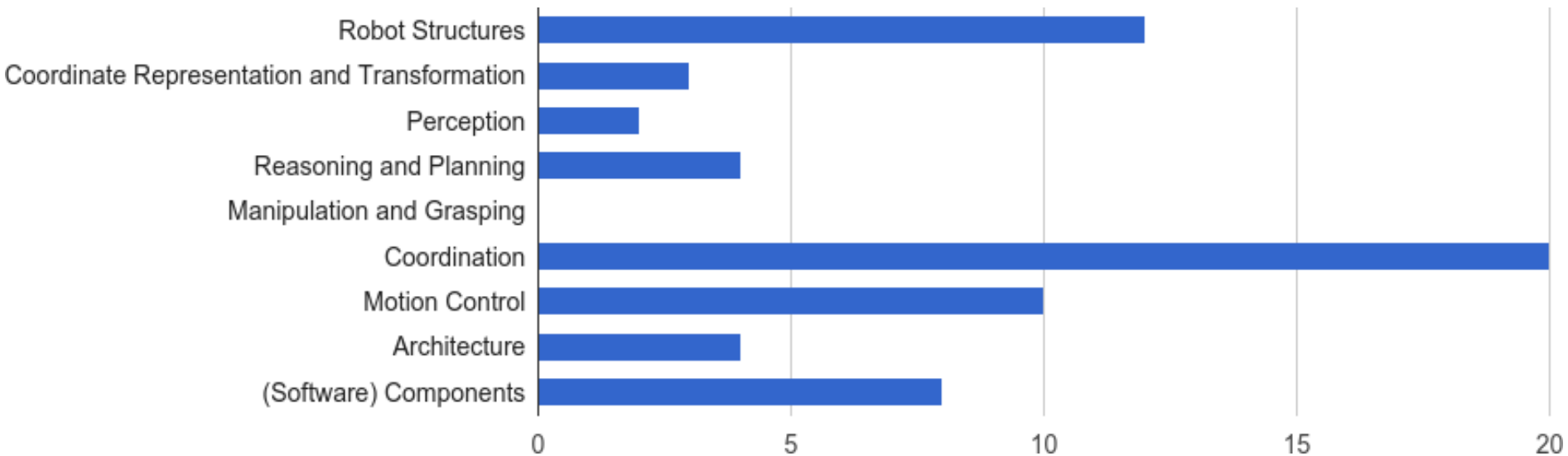
Survey: Publications per Year

1. Numbers clearly support a positive trend of DSLs in robotics
2. Numerous publications per year since 2010 (DSLRob start)



Survey: DSLs per Subdomain

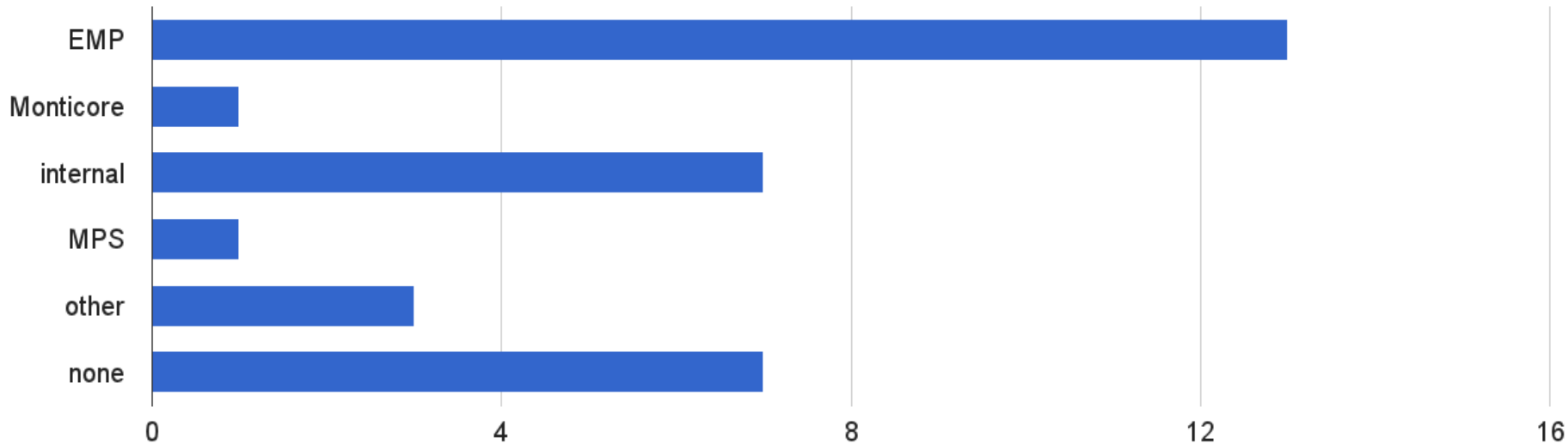
1. Numbers vary significantly between subdomains
2. Task-level coordination well-explored (> 20, mature?)
3. Robot Structure and Motion Control > 10
4. Manipulation and Grasping none



Indicator for maturity of the domain?

Survey: DSLs per Tool

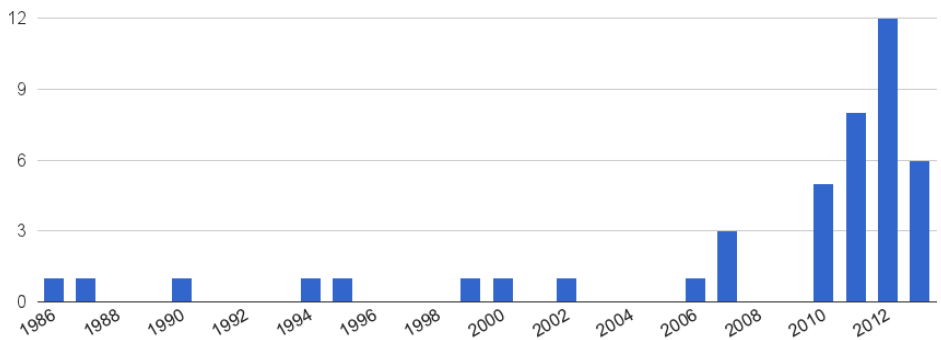
1. Eclipse Modeling Project (including EMF, GMF, xtext, xpand, . . .) leading by far



Is Eclipse Modeling Project a potential integration point for robotics DSLs?

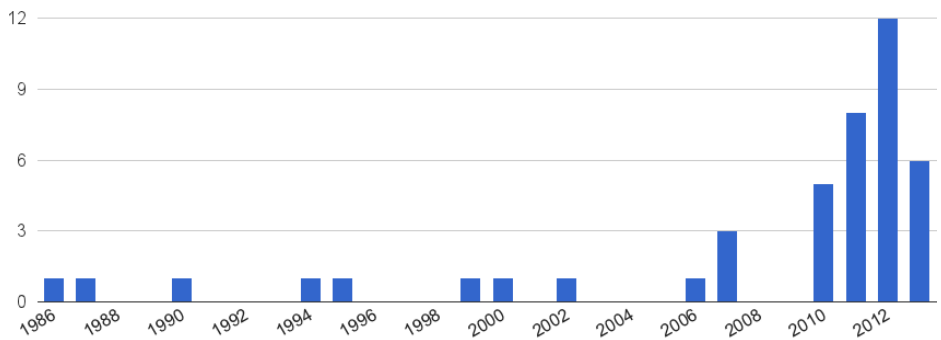
Survey Summary

1. Started DSL survey with 6 robotics conferences and 2 software conferences [3]
2. Filtered by robotics focus and technical aspects
3. $\Rightarrow$ **41 publications** *(will be continued)*
4. Preliminary results
   - Supports positive trend for DSLs in robotics
   - Varying DSL support for subdomains
   - EMF possible integration point
   - Identified further relevant aspects of DSLs in robotics



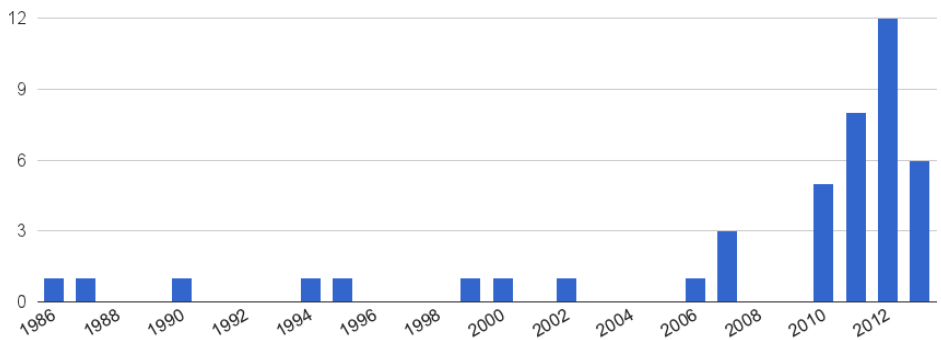
Survey Summary

1. Started DSL survey with 6 robotics conferences and 2 software conferences [3]
2. Filtered by robotics focus and technical aspects
3. $\Rightarrow$ **41 publications** *(will be continued)*
4. Preliminary results
   - Supports positive trend for DSLs in robotics
   - Varying DSL support for subdomains
   - EMF possible integration point
   - Identified further relevant aspects of DSLs in robotics



Survey Summary

1. Started DSL survey with 6 robotics conferences and 2 software conferences [3]
2. Filtered by robotics focus and technical aspects
3. $\Rightarrow$ **41 publications** *(will be continued)*
4. Preliminary results
   - Supports positive trend for DSLs in robotics
   - Varying DSL support for subdomains
   - EMF possible integration point
   - Identified further relevant aspects of DSLs in robotics



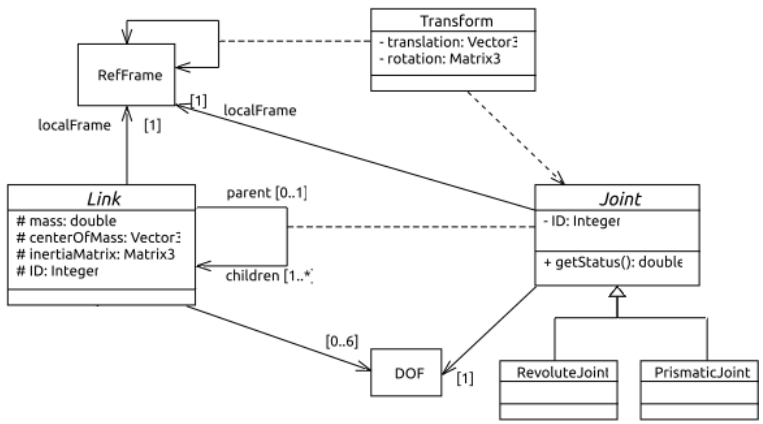
Discussion

Relevant aspects of DSLs from developer and user perspective

Accessibility and Documentation

Documentation necessary for re-use of DSLs, scientific exchange and community building.

1. Technical accessibility
   - Download of DSL
   - Download of language models
   - Download of tools
2. Licensing
3. Documentation of DSL usage, examples, tutorials, ...

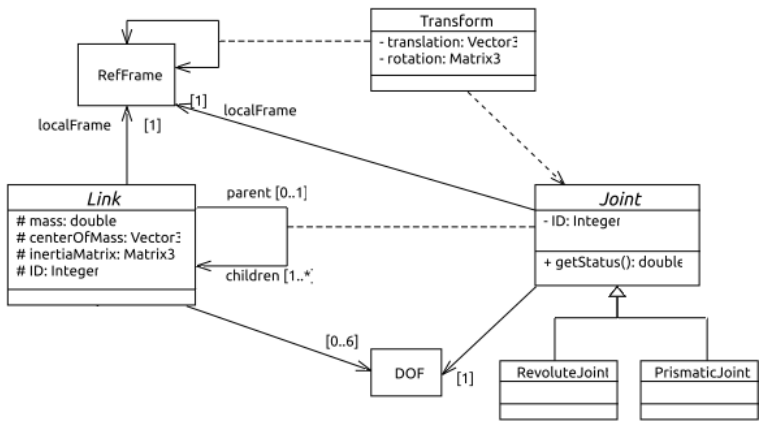


Best Practice: *Documentation of meta-model (Ecore, EBNF, ...), intended use-case (tutorial?), open-source download.*

Accessibility and Documentation

Documentation necessary for re-use of DSLs, scientific exchange and community building.

1. Technical accessibility
   - Download of DSL
   - Download of language models
   - Download of tools
2. Licensing
3. Documentation of DSL usage, examples, tutorials, ...



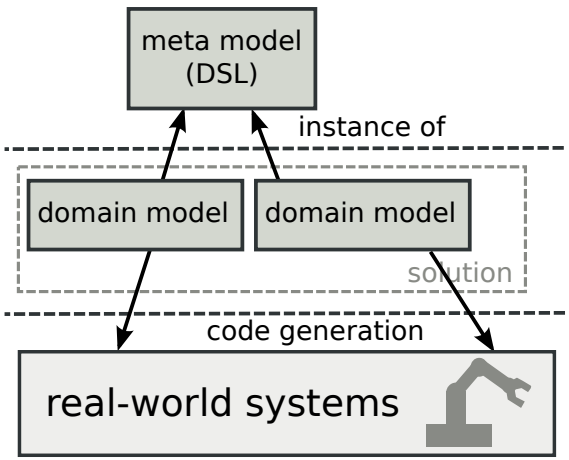
Best Practice: *Documentation of meta-model (Ecore, EBNF, ...), intended use-case (tutorial?), open-source download.*

Artifacts and Use-Case

1. DSLs of same sub-domain usually generate similar artifacts
2. Primary target: generation of executable code, but also documentation and visualization
3. Most DSLs generate *one* artifact type (e.g. C++ code)

Generation becomes more powerful when parallel M2M/M2T generators are supported, e.g.

- Computational code and glue code
- For different software platforms
- For different programming languages



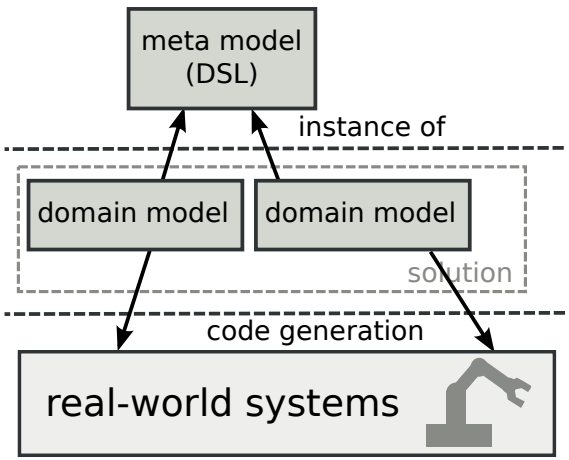
Best Practice: *Support for parallel generators (M2M and M2T) eases DSL re-use.*

Artifacts and Use-Case

1. DSLs of same sub-domain usually generate similar artifacts
2. Primary target: generation of executable code, but also documentation and visualization
3. Most DSLs generate *one* artifact type (e.g. C++ code)

Generation becomes more powerful when parallel M2M/M2T generators are supported, e.g.

- Computational code and glue code
- For different software platforms
- For different programming languages



Best Practice: *Support for parallel generators (M2M and M2T) eases DSL re-use.*

Platform

“Coupling” – All tools and software libraries required to use the DSL or the generated artifacts.

1. Interpreted / executed DSLs:
   - Always coupled with (DSL-specific) interpreter
2. DSLs for code generation:
   1. Proprietary (KRL, RAPID, . . .) strongly tied to one platform
   2. Tied to a certain software library stack or tool
   3. General purpose language code without dependencies

Best Practice: *Less platform dependencies ease DSL re-use.*

Platform

“Coupling” – All tools and software libraries required to use the DSL or the generated artifacts.

1. Interpreted / executed DSLs:
   - Always coupled with (DSL-specific) interpreter
2. DSLs for code generation:
   1. Proprietary (KRL, RAPID, . . .) strongly tied to one platform
   2. Tied to a certain software library stack or tool
   3. General purpose language code without dependencies

Best Practice: *Less platform dependencies ease DSL re-use.*

Evaluation

Quality/usability of the DSL or its generated artifacts.

1. Qualitative evaluation
   - Discussion within a use-case (simulation or hardware)
   - Suitability, portability, ...
   - Surprising number of publications evaluated on hardware (even on different robot platforms)
2. Quantitative evaluation
   - Efficiency (benchmarking, computation time, ...)
   - Scalability (compilation time, system size, ...)
   - Productivity (change requests, dev time, ...)
   - Reliability (errors/defects per time, ...)

Best Practice: *Qualitative evaluation in multiple use-cases and quantitative evaluation.*

Evaluation

Quality/usability of the DSL or its generated artifacts.

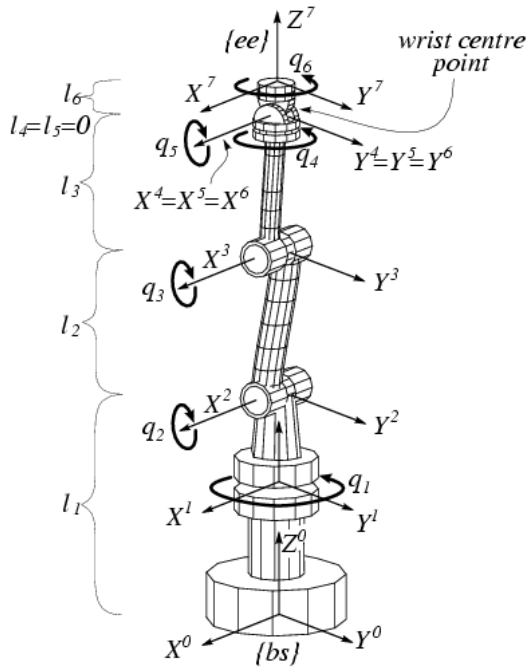
1. Qualitative evaluation
   - Discussion within a use-case (simulation or hardware)
   - Suitability, portability, ...
   - Surprising number of publications evaluated on hardware (even on different robot platforms)
2. Quantitative evaluation
   - Efficiency (benchmarking, computation time, ...)
   - Scalability (compilation time, system size, ...)
   - Productivity (change requests, dev time, ...)
   - Reliability (errors/defects per time, ...)

Best Practice: *Qualitative evaluation in multiple use-cases and quantitative evaluation.*

DSL Development Process

Process of identification and formalization of domain-specific abstractions

1. Adds credibility to the DSL
2. Only little information available
   - Domain analysis?
   - Ontology?
   - Formalism?
   - Software examples?
   - Handbooks?

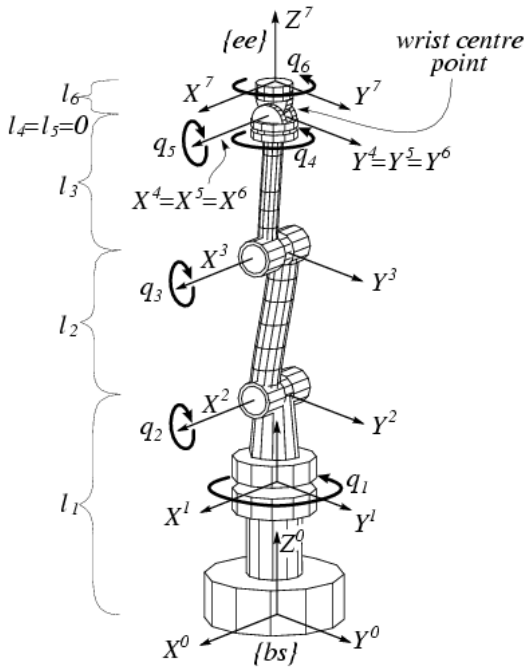


Best Practice: *Link/describe sources of your DSL: domain analysis, ontology, formalism, . . .*

DSL Development Process

Process of identification and formalization of domain-specific abstractions

1. Adds credibility to the DSL
2. Only little information available
   - Domain analysis?
   - Ontology?
   - Formalism?
   - Software examples?
   - Handbooks?



Best Practice: *Link/describe sources of your DSL: domain analysis, ontology, formalism, . . .*

Robotics DSL Zoo

Online resource for DSLs in robotics

Robotics DSL Zoo

Online resource to structure, consolidate and harmonize DSL developments in robotics.

- Provide a **DSL "map"** for (potential) DSL users
- **Foster scientific exchange** and community building

Structure:

1. DSL Collection, inspired by the **EMFText Concrete Syntax Zoo**[1]
2. Annotated bibliography, inspired by Van Deursen et al.[2]

http://cor-lab.org/robotics-dsl-zoo

[1] http://www.emftext.org/index.php/EMFText_Concrete_Syntax_Zoo

[2] A. van Deursen, P. Klint, and J. Visser: Domain-Specific Languages: An Annotated Bibliography, 2000.

Robotics DSL Zoo – Contribute

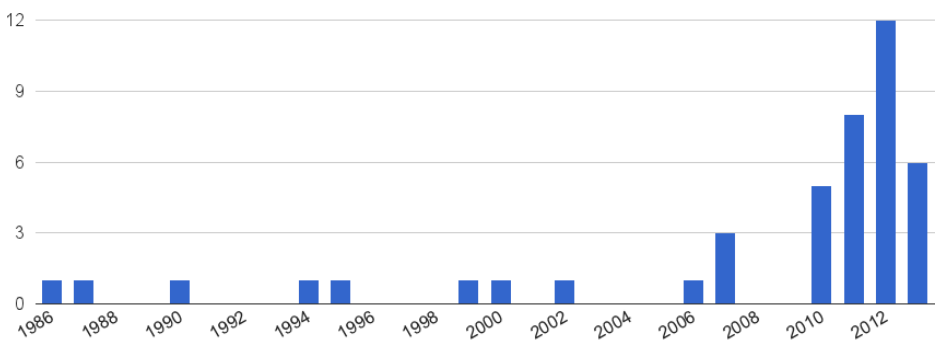
We invite the Robotics DSL Community to contribute.

1. Find a DSL you know / use / develop
2. Targets relevant concern for robotics?
3. Collect metadata (publication, authors, year)
4. Assess:
   1. Documentation and accessibility (website, download, metamodel, . . .)
   2. Artifacts and use-case (e.g. controller configuration, . . .)
   3. Platform dependencies (required tools and software library . . .)
   4. Evaluation (in simulation, on hardware, on different platforms, . . .)
   5. Sources of DSL development (domain analysis, ontology, . . .)

http://cor-lab.org/robotics-dsl-zoo

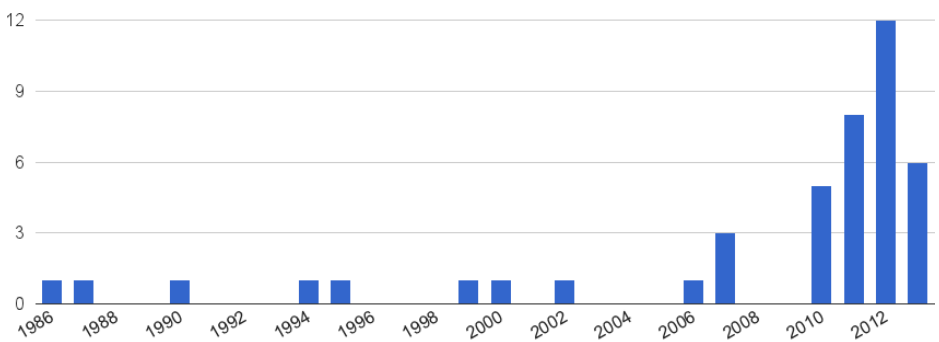
Conclusion

1. **DSLs are on the rise in robotics!** [3] *Yay!*
2. Community building started, but still **lack of accessibility, documentation and exchange**
   - Technical accessibility (download)
   - Meta-model documentation and download
   - Use-cases and tutorials
3. **Robotics DSL Zoo** is our idea to ...
   - ... **structure, consolidate and harmonize** domain-specific language developments in robotics
   - ... provide a **DSL "map"** for (potential) DSL users
   - ... **foster scientific exchange** and community building



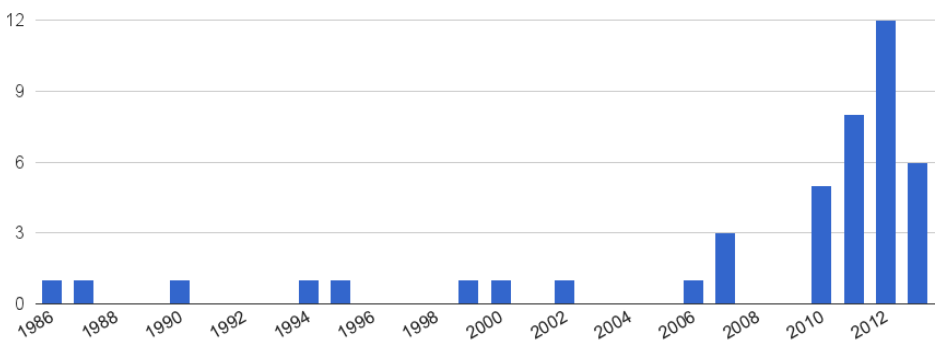
Conclusion

1. **DSLs are on the rise in robotics!** [3] *Yay!*
2. Community building started, but still **lack of accessibility, documentation and exchange**
   - Technical accessibility (download)
   - Meta-model documentation and download
   - Use-cases and tutorials
3. **Robotics DSL Zoo** is our idea to . . .
   - . . . **structure, consolidate and harmonize** domain-specific language developments in robotics
   - . . . provide a **DSL "map"** for (potential) DSL users
   - . . . **foster scientific exchange** and community building



Conclusion

1. **DSLs are on the rise in robotics!** [3] *Yay!*
2. Community building started, but still **lack of accessibility, documentation and exchange**
   - Technical accessibility (download)
   - Meta-model documentation and download
   - Use-cases and tutorials
3. **Robotics DSL Zoo** is our idea to ...
   - ... **structure, consolidate and harmonize** domain-specific language developments in robotics
   - ... provide a **DSL "map"** for (potential) DSL users
   - ... **foster scientific exchange** and community building



Thank you for your attention!

Arne Nordmann, Sebastian Wrede, Bielefeld University
{*anordman, swrede*}*@cor-lab.uni-bielefeld.de*

Nico Hochgeschwender, Bonn-Rhein-Sieg University
nico.hochgeschwender@h-brs.de

http://cor-lab.org/robotics-dsl-zoo

CORLab
Research Institute
for Cognition and Robotics

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Universität Bielefeld

References


Markus Völter, Sebastian Benz, Christian Dietrich, Birgit Engelmann, Mats Helander, Lennart Kats, Eelco Visser, and Guido Wachsmuth.
DSL Engineering Designing, Implementing and Using Domain-Specific Languages.
2013.


M. Frigerio, J. Buchli, and D.G. Caldwell.
A Domain Specific Language for Kinematic Models and Fast Implementations of Robot Dynamics Algorithms.
In *Workshop on Domain-Specific Languages and models for Robotic systems*, 2011.


Arne Nordmann, Nico Hochgeschwender, and Sebastian Wrede.
A survey on domain-specific languages in robotics.
International Conference on Simulation, Modeling, and Programming for Autonomous Robots, 2014.