

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

Statemer

JPLS IPLS

LWs

LSi

The Reductions
of Combination

Contributio

Type System Components

LSP in Action

Toward a Modular Approach for Type Systems and LSP generation

Federico Bruzzone

ld. Number: 27427A

Università degli Studi di Milano Computer Science Department MSc in Computer Science

Advisor: Prof. Walter Cazzola

Co-Advisor: Dr. Luca Favalli

15/07/2024

LM-18 - Computer science Academic Year 2023-2024





Problem Statement Programming Language Implementation

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

The implementation of a programming language is a complex task that involves several implementation aspects, such as:

- Syntax and semantics definition
- Type system definition
- Code generation

- Error handling and recovery
- IDE support
- Documentation





Problem Statement Programming Language Implementation

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

The implementation of a programming language is a complex task that involves several implementation aspects, such as:

- Syntax and semantics definition
- Type system definition
- Code generation

- Error handling and recovery
- IDE support
- Documentation

It is usually done in a monolithic way, where all the aspects are tightly coupled.





Problem Statement Programming Language Implementation

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

The implementation of a programming language is a complex task that involves several implementation aspects, such as:

- Syntax and semantics definition
- Type system definition
- Code generation

- Error handling and recovery
- IDE support
- Documentation

It is usually done in a monolithic way, where all the aspects are tightly coupled.

This makes the maintainability, extensibility and reusability of the implementation difficult.





Problem Statement Type Systems and IDEs Support

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

SPLs LPLs

LPLs LWs

The Reducti

In a Nutshell

Contribution

Type System Components

LSP in Action

Often some parts of compilation, such as code generation, makes use of feature-oriented programming to support different architectures.





Problem Statement Type Systems and IDEs Support

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemer

SPLs LPLs

The Reduct

of Combination In a Nutshell

Contribution

Type System Components

LSP in Action

Often some parts of compilation, such as code generation, makes use of feature-oriented programming to support different architectures.

However, the type system and the IDE support are usually implemented using a top-down approach.





Software Product Lines

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemer

Stateme

LPLs LWs

LSF

of Combination

Contributio

Type System

LSP in Action

Since 1990s, researchers have been working on the concept of Software Product Lines (SPLs) to move towards a more modular world.





Software Product Lines

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Stateme

SPLs LPLs

LWs ISP

The Reduction of Combinatio

of Combinations In a Nutshell

Type System Components

LSP in Action

Since 1990s, researchers have Been working on the concept of Software Product Lines (SPLs) to move towards a more modular world.

- SPLs defines a family of software products.
- SPLs is described by a Feature Model.
- A Feature Model describes the variability of the software.
- SPL variants are generated by selecting a set of features.
- A feature (or artifact) is a first-class entity in SPLs.





Software Product Lines Language Product Lines

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

Stateme

SPL

LPL

LS

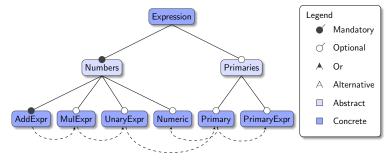
of Combinat

Contribution

Type System Components

LSP in Action

Applying the concept of SPLs to programming languages, we obtain the concept of Language Product Lines (LPLs).







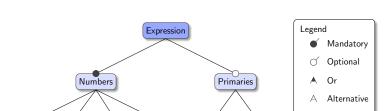
Toward a

Approach for TSs and LSP generation Federico

Bruzzone

Software Product Lines Language Product Lines

Applying the concept of SPLs to programming languages, we obtain the concept of Language Product Lines (LPLs).



Primary

PrimaryExpr

Some achievements:

MulExpr

AddExpr

- Bottom-up approach to language implementation

Numeric

- Reusability of language artifacts
- Multiple variants of the same language

UnaryExpr

- Language Worksenches come to the rescue



Abstract

Concrete

Slide 5 of 10



Software Product Lines Language Workbenches and Neverlang

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemen

SDIC

LPLs

LWs

The Redu

of Combination

CONTRIBUTIO

Components

LSP in Action

Language Workbenches (LWs) are tools that allow the development of programming languages, Both GPLs and DSLs. Some LWs allow the development of LPLs.

Language Workbench	Modularization Supp.	Precompiled Feature Supp.	Native IDE gen.	LSP Gen	LSP Mod.
JustAdd	0	0	0	0	0
Melange	0	0	2rd party (EMF)	्रो	्रेट
MontiCore	0	0	•	0	0
MPS	0	0	•	sk.	sh.
Rascal	0	0	•	0	0
Spoofax	0	0	•	St.	sh.
Ytext	0	0	•	•	0
Neverlang	⊘	•	0	*	*

- Full support
- O No support
- Limited support
- O Fine-grained mod.

- O Coarse-grained mod.
- ★ My contribution
- ☆ Future Work





Software Product Lines Language Workbenches and Neverlang

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemen

SPLs

LPLs LWs

LWs

The Reduc of Combir

In a Nutshell

T- Color

Type System Components

LSP in Action

Language Workbenches (LWs) are tools that allow the development of programming languages, Both GPLs and DSLs. Some LWs allow the development of LPLs.

anguage Workbench	Modularization Supp.	Precompiled Feature Supp.	Native IDE gen.	LSP Gen	LSP Mod
JustAdd	0	0	0	0	0
Melange	0	0	2rd party (EMF)	清	shr.
MontiCore	•	0	•	0	0
MPS	0	0	•	St.	St.
R.ascal	0	0	•	0	0
Spoofax	0	•	•	St.	St.
Ytext	0	•	•	•	0
Neverlang	©	•	0	*	*

- Full support
- O No support
- O Limited support
- @ Fine-grained mod.

- O Coarse-Grained mod.
- ★ My contribution
- ☆ Future Work

Neverlang is a language workbench, developed by the ADAPT lab, that supports the development of LPLs.



Language Server Protocol The Reduction of Combinations

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

Statemer

SPLS LPLS

LWs

LSi

The Reductions of Combinations

Contribution

Type System Components

LSP in Action

In 2016, Microsoft in collaboration with Red Hat introduced the Language Server Protocol (LSP).





Language Server Protocol The Reduction of Combinations

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemer

Stateme

SPLS LPLS

LW:

The Reductions

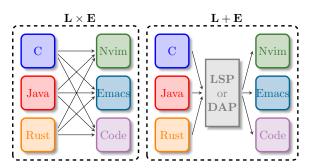
of Combinations In a Nutskell

_ -

Type System Components

LSP in Action

In 2016, Microsoft in collaboration with Red Hat introduced the Language Server Protocol (LSP).







Language Server Protocol The Reduction of Combinations

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

Stateme

LPLs LWs

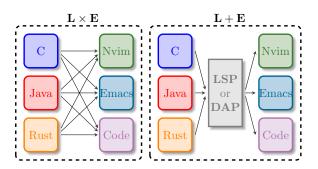
LSF

The Reductions of Combinations In a Nutshell

Type System

LSP in Action

In 2016, Microsoft in collaboration with Red Hat introduced the Language Server Protocol (LSP).



Spoiler: We have reduced the number of combinations from $L \times E$ to $N \times 1$ where $N \ll L$



Language Server Protocol LSP In a Nutshell

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

In a Nutshell

The Language Server Protocol (LSP) is a protocol that allows the communication between a Language Server and an IDE.





Language Server Protocol LSP In a Nutshell

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemer

Stateme

SPL:

LWs

The

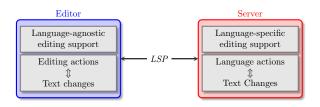
of Combinat

Contracto

Type System

LSP in Action

The Language Server Protocol (LSP) is a protocol that allows the communication between a Language Server and an IDE.



Intrinsic properties:

- Language-agnostic
- IDE-agnostic
- Asynchronous
- Text-Based

Features:

- Diagnostics
- Hover
- Go to definition
- Find references
- Inlay hints





Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem

Stateme

SPLs

LPLS

LSF

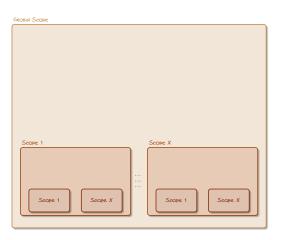
The Reduction of Combinati

In a Nuts

CONTRIBUTION

Type System Components

SP in Action







Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

COL -

LWs

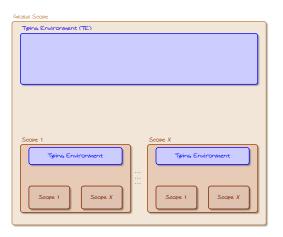
LSF

of Combinati

IN a Nurtsi

Type System Components

LSP in Action



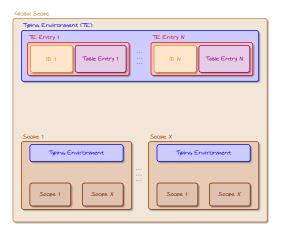




Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Type System Components







Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem

Stateme

SPLs

LWs

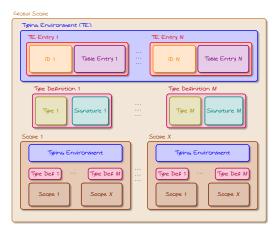
LSF

The Reduction of Combinati

in a Nutshell

Type System Components

SP in Action







Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem

Stateme

SPLs

LPLS

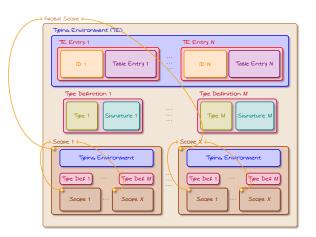
LSi

The Reduction of Combinat

IN a Nuclsine

Type System Components

LSP in Action







Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem Statemen

SPLs

1105

LSF

The Reductions of Combinations

IN a NUCSNEII

Type System

LSP in Action

