

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemen

Statemer

SPLS

LWs

LSP

The Reductions of Combinatio

IT B TAU CSTYCII

Contribution

Type System Components

Type System Modularizatio

SP 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

task that involves several implementation aspects, such as:

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

- Error handling and recovery
- IDE support

The implementation of a programming language is a complex

- 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

Problem Statemer

STATEME

LPLs

LWs

The

The Reductions of Combinatio In a Nurtshell

Contributio

Components
Type System

P in Actior

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.

The implementation of a programming language is a complex

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 R

of Combination

Contribution

Type System Components

Type System Modularizatio

SP in Actio

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

LSP

of Combinatio

Contribution

Type System Components

Modularizati

SP in Actic

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

SOL

LPLs

LSP

of Combinatio

Contribution

Type System Components

Modularizatio

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

The Reduction

of Combinations In a Nutshell

Type System
Components
Type System
Modularizatio

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.



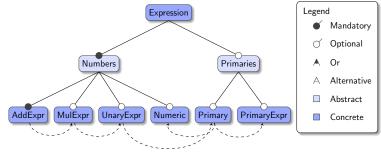


#### Software Product Lines Language Product Lines

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

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







### Software Product Lines Language Product Lines

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemen

---

SPLS

LPLS

1 <

The Reduct of Combine

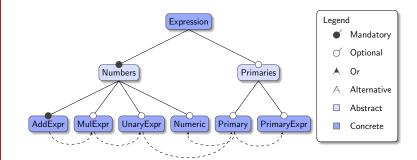
Contribution

Type System Components

160: 4 ::

LSP in Action

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



### Some achievements:

- Bottom-up approach to language implementation
- Reusability of language artifacts
- Multiple variants of the same language
- Language Workbenches come to the rescue





### Software Product Lines Language Workbenches and Neverlang

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Statemen

SPLs

LWs

The R

of Combinati In a Nutshell

Contribution

Components
Type System

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 Worksench	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	St.
MontiCore	0	0	•	0	0
MPS	0	0	•	St.	<b>\$</b>
Rascal	0	0	•	0	0
Spoofax	0	0	•	St.	shr.
Ytext	0	0	•	•	0
Neverland	<b>Q</b>	•	0	*	*

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

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





### Software Product Lines

Language Workbenches and Neverlang

INJORGERANDIAS (1)

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Statemen

SPLs

LPLs LWs

The Reduc

of Combinati In a Nutshell

Contribution

Components

....

.SP in Action

Lor Iquoge	vool ree iches	(CVV3)	016	COOIS	that	allOW	ine
developmen	t of programm	ning land	uages	s, Both	GPLs	and DS	Ls.
Some LWs allow	u the development of L	PLs.					

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)	्रो	्री	
MontiCore	•	•	•	0	0	
MPS	0	0	•	sk.	St.	
R.ascal	0	0	•	0	0	
Spoofax	0	•	•	sk.	St.	
Ytext	0	•	•	•	0	
Neverlang	©	•	0	*	*	

- Full support
- O No support
- 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

The Reductions of Consinations

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

SPLS

LWs

LSi

The Reductions of Combinations In a Nutshell

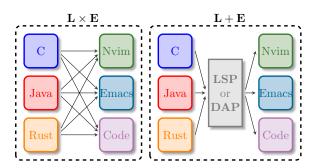
COME IBUTE

Type System Components

. 100000 120000

SP in Action

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







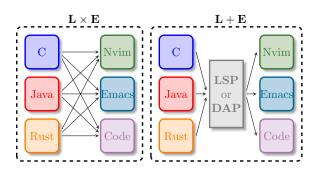
#### Language Server Protocol The Reduction of Combinations

Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

The O educations of Consinations

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

SDI 2

1014

LW

LSi

of Combinat

#### ...

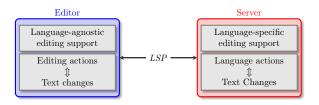
\_ -

Components

Modular Ization

SP 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 Statemen

Stateme

SPLS

LW:

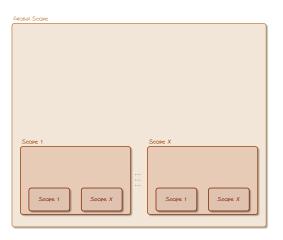
LSi

of Combinati

Type System Components

Type System

.SP in Actio







Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemer

Stateme

SPLs

LWs

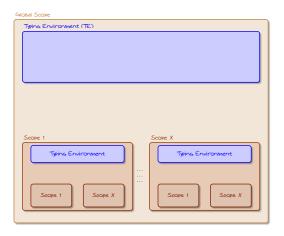
LSF

of Combinati

Type System Components

Type System

.SP in Action







Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem Statemer

Stateme

SPLs

LWs

LSF

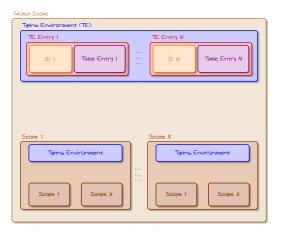
The Reduction of Combinati

in a Nurtske

Type System Components

Type System

LSP in Action



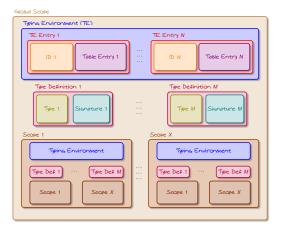




Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Type System Components







Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Problem

Stateme

SPLs

LWs

LSi

The Reduction of Combination

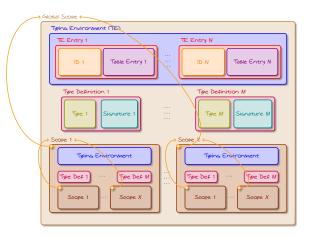
IT B TAKESTY

CONTRIBUTIO

Type System Components

Type System Modularizati

SP in Action







# Contribution Type System Modularization

Toward a Modular Approach for TSs and LSP Generation

Federico Bruzzone

Problem

Statemer

SPLs

LWS

LSi

The Reduction of Combinat

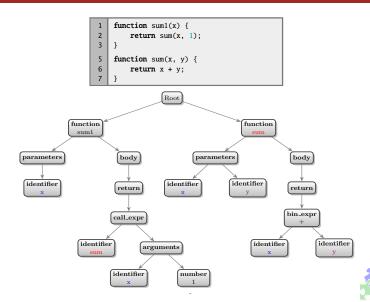
In a Nut

COME IBUTE

Component

Type System Modularization

SP in Action





#### Contribution Type System Modularization

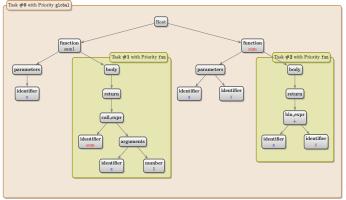
Toward a Modular Approach for TSs and LSP generation

Federico Bruzzone

Type System Modularization

```
function sum1(x) {
       return sum(x. 1):
2
   function sum(x, y) {
6
        return x + y;
```

- Compilation Unit
- Compilation Unit Task
- Compilation Helper







Toward a Modular Approach for TSs and LSP generation Federico

Bruzzone

Problem Statemer

SPLs

LPLs 1 \n/s

LSP

The Reductions of Combinations

Type System

Type System

LSP in Action



Toward a Modular Approach for TSs and LSP generation

> Federico Bruzzone

Problem Statemen

SPLs

JPLS

1 00

LSP

The Reductions of Combination

In a Nutsi

Contribution

Type System

Type System

LSP in Action

### Thanks for your attention!

