

Universal LSP and DAP for Modular LWs

Eederico Bruzzone

Universal Language Server Protocol and Debugger Adapter Protocol for Modular Language Workbenches

Federico Bruzzone

Università degli Studi di Milano Computer Science Department PhD Candidate in Computer Science

> 22/07/2024 Cyclus 40th





Problem Statement Programming Language Implementation

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statement

Statemen

In a Nutshell

The Reductions of Combination

EOP

LW

Scientific Contribution

Type System Component

Mad Jaria adia

Condusions

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
- IDE support
- Documentation





Problem Statement Programming Language Implementation

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statement

Statemen

In a Nutshell

The Reductions
of Combination

FOF

Sojenti Sin

Scientific Contributio

Components Checking an

Modularization

Conclusions

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
- IDE support
- Documentation

It is usually done in a monolithic way with a top-down approach, where all the aspects are tightly coupled.





Problem Statement Programming Language Implementation

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statemen

LSP +DA

In a Nietsk

The Reductions of Combination

An Achieveme

FOF

Scientific

Contributio

Checking an Inference

Modularization

Conclusions

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
- IDE support
- Documentation

It is usually done in a monolithic way with a top-down approach, where all the aspects are tightly coupled.

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





LSP and DAP In a Nutshell

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statemer

1 SD EDA

In a Nutshell

The Reductions of Combinatio

EQ5

LWS

Scientific Contribution

Components

Inference

Modularization

Conclusions

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





LSP and DAP In a Nutshell

Universal LSP and DAP for Modular LWs

Eederico Bruzzone

In a Nutshell

In 2016. Microsoft in collaboration with Red Hat introduced the Language Server Protocol (LSP) and the Debugger Adapter Protocol (DAP)

The LSP and DAP are JSON-RPC Based protocols that allow the communication between a Language Server and an IDE.





LSP and DAP In a Nutshell

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Statemer

LSP &DAF

In a Nutshell
The Reduction

of Comsinatio
An Achievemen

FOF

Contributio

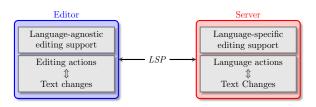
Checking ar

Modularization

Conclusion

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

The LSP and DAP are JSON-RPC Based protocols that allow 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





LSP and DAP The Reduction of Combinations

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Stateme

LSP +DAI

In a Nutske

The Reductions of Combinations

FOF

LWs

Scientific Contribution

Component

Madded

Modularization

Conclusions

Initially implemented for Visual Studio Code, the LSP and DAP have been adopted by several IDEs and programming languages.





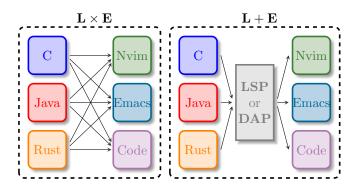
LSP and DAP The Reduction of Combinations

Universal LSP and DAP for Modular LWs

Eederico Bruzzone

The Reductions of Combinations

Initially implemented for Visual Studio Code, the LSP and DAP have been adopted by several IDEs and programming languages.







Federico Bruzzone

LSP and DAP

What would be an important achievement?

Universal LSP Reducing the number of combinations between Language and DAP for Servers and IDEs.

Problem Statem

Stateme

LOP TUAL

In a Nutshell

of Combinati

An Achievement

EU5

Salandidia

Scientific Contribution

Components Checking and

Modularization

Modularization

Conclusion





LSP and DAP What would be an important achievement?

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statemer

150404

LSP FDA

The Reduction

of Combination

An Achievement

FOF

_ .

Scientific Contribution

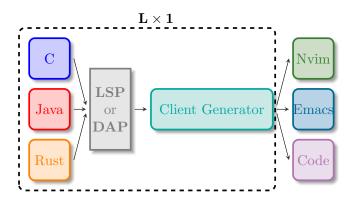
Component Checking an

Modularization

Canalysians

Condusio

Reducing the number of combinations between Language Servers and IDEs.



RO 3: Reduce to $L \times 1$ the number of combinations to support L languages

Slide 5 of 14



Feature-Oriented Programming

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem

150 4740

In a Nutrical

The Reductions of Combinatio

FOR

C = 1 = - 41 01 =

Scientific Contribution

Component

Maduariantia

Modularization

Feature-Oriented Programming (FOP) is a programming paradigm that allows the development of software product lines (SPLs).





Feature-Oriented Programming

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Statemen

LSP +DAP

In a Nutshell The Reduction of Combination

An Achieveme

FOP

Scientific

Contribution

Components
Checking and

Modularizatio

Conclusions

Feature-Oriented Programming (FOP) is a programming paradigm that allows the development of software product lines (SPLs).

- Feature is a unit of functionality that satisfies a requirement.
- Feature Model is a model that represents the variability of the SPL.
- Feature Configuration is a set of features that compose a product.

RO 2: Facilitate LSP and DAP Modularization





Feature-Oriented Programming

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Statemen

1 SD &DA1

LSP FDA

The Reduction

An Achievem

FOP

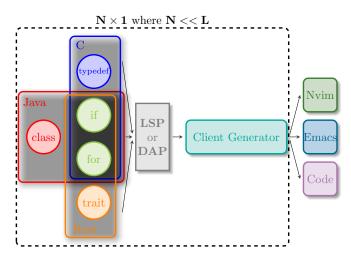
Scientific Contribution

Components

AA-10-co

Modularization

Conclusion







Language Workbenches

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Statemer

LSP +DA

The Reductions

of Combinatio

LWs

Scientific

Type System

Components
Checking an

Modularizatio

Conclusion

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

Language Workbench	Modularization Supp.	Precompiled Feature Supp.	Native IDE gen	LSP & DAP Gen	LSP ≠ DAP Mod.
JustAdd	•	0	0	0	0
Melange	0	0	3rd p.	1,7	\$
MontiCore	•	•	•	0	0
MPS	0	0	•	☆	☆
Rascal	0	0	•	0	0
Spoofax	0	•	•	☆	☆
Ytext	0	•	•	•	0
Neverlang	©	•	0	*	☆

- Full support
- No support
- $lackbox{ Limited support}$
- Fine-grained mod.

- O Coarse-Grained mod.
- My expected contribution
- ☆ Extended contribution

3rd p. Third-party

RO 1: Improve IDE and LSP Generation





Scientific Contribution

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Stateme

LSP +DA

In a Nutshell

The Reductions
of Combination

FOF

Saiandi Nia

Scientific Contribution

Components
Checking and
Inference

Conclusio

 Methodology for whole LWs that support at least component modularization.

- Type System, LSP and DAP Modularization.
- DSL for Type System definition.
- LSP and DAP generation for Neverlang languages.
- Clients and Syntax Highlighting generation reducing the number of combinations.
- Implementation of a Java Library for Neverlang to support the type system, LSP and DAP for every language developed with Neverlang.
- 3 use cases to show the effectiveness of the methodology.

RO 4: Leverage Neverlang for LSP and DAP in LPL Development



Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem

LSP +DAF

In a Nutshe

The Reduction

of Combinatio

COS

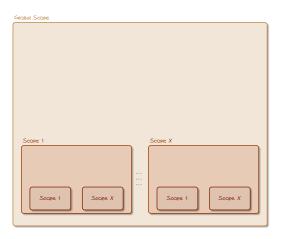
Contribution
Type System

Components

interence

Modular Izartion

Conclusion







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem

LSP +DA

LSP TUA

The Reduction

of Combination

COS

2003

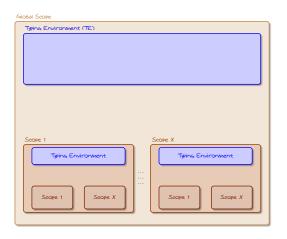
Scientific Contribution

Type System Components

Inference

Modularization

Londusion







Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Stateme

1 SD & DAG

LSP &DA

The Reduction

of Combinatio

COS

2003

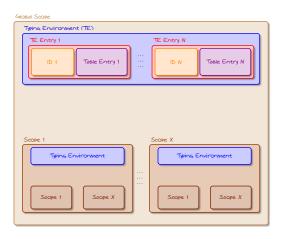
Scientific Contribution

Type System Components

Inference

Modularization

-onelusion







Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Stateme

1 <D &D A1

LSP FDA

The Reductio

of Combinatio

FOR

L 003

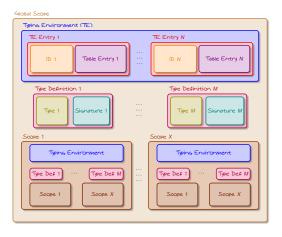
Scientific Contributio

Type System Components

Inference

Modularization

Conclusions







Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem

Stateme

LSP +DA

The Reduction of Combination

An Achievem

FOP

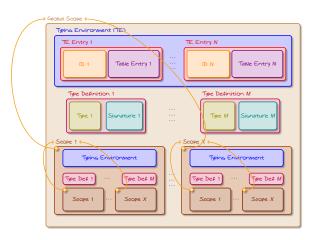
Scientific Contributio

Type System Components

Inference

Modularization

Conclusions







Scientific Contribution Type Checking and Type Inference

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Stateme

Stateme

LSP &DA

In a Niertska

The Reductions

AN A

-01

Scientific

Type System

Checking and Inference

Modularization

Conclusions

function sum1(x) { 1 2 return sum(x. 1): 3 5 function sum(x, y) { 6 return x + y; 7 Root function function sum1 sum parameters body parameters body identifier identifier identifier return return bin_expr call_expr identifier identifier identifier arguments sum identifier number



Scientific Contribution Type Checking and Type Inference

Universal LSP and DAP for Modular LWs

Federico Bruzzone

Problem Stateme

LSP *DA

The Reductions of Combination

LW:

Scientific Contributio

Components
Checking and

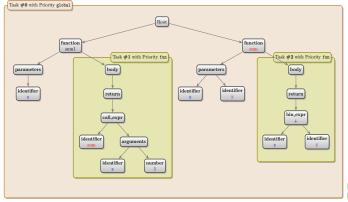
Modularization

. . .

Conclusions

```
1 function sum1(x) {
2    return sum(x, 1);
3  }
5 function sum(x, y) {
6    return x + y;
7  }
```

- Compilation Unit
- Compilation Unit Task
- Compilation Helper







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

150 671

LSP &DA

The Reduction

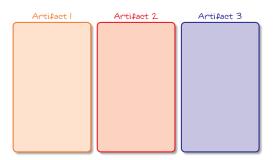
of Combinatio

COP

Scientific Contribution

Component

Modularization







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

I SO EDAT

LSP *DAI

The Deduction

of Combinatio

COS

LW:

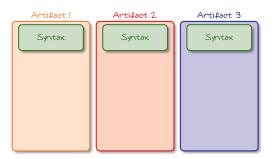
Scientific

Component

Modularization

Modularization

Conclusions







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

LSP +DA

In a Milatel

The Reduction

of Combinatio

FOR

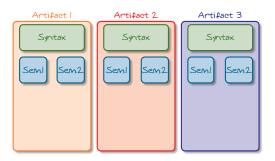
2003

Scientific Contribution

Components

Modularization

Conducione







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

Stateme

In a Nuts

The Reductions of Combinatio

An Achieven

FOP

C = 1 = = = #1 0

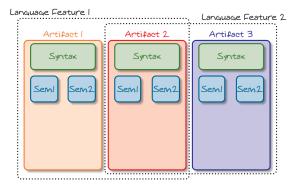
Scientific Contributio

Components

Modularization

Modularization

Condusions







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

LSP *DP

The Reduction of Combination

An Ach

FOP

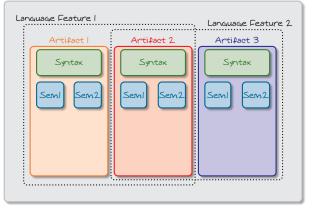
_ .

Contributio

Components

Modularization

Conclusions







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

In a Nurtel

The Reductions of Combinatio

An Achieveme

FOP

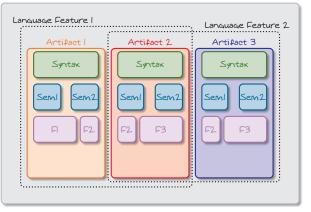
_ .

Scientific Contributio

Component

Modularization

Conclusion







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

LOP #UP

The Reduction of Combinatio

An Achievem

FOP

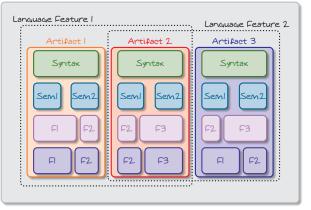
_ .

Scientific Contribution

Component Checking an

Modularization

Conclusion



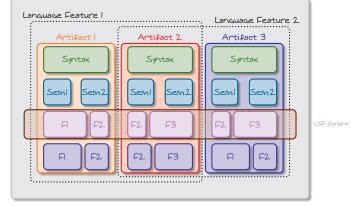




Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Modularization







Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

LSP +DA

In a Nutsh

The Reduction of Combinatio

FOF

LVV

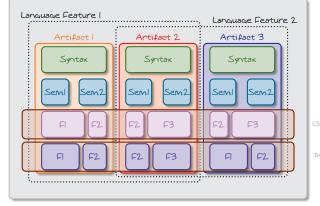
Scientific Contribution

Components
Checking an

Modularization

Condusion





LSP Variant {Feature 1 Feature 2

DAP Variant Feature 1

Feature 3





Conclusions Master's Thesis Results

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Statem

LSP #DAP
In a Nutshell
The Reductions

The Reductions of Combination An Achievement

FOF

Scientific

Contribution
Type System

Checking and Inference

Modularization

Condusions

Interesting results:

- We are writing an article (Code Less to Code More) to be submitted to JSS.
- Propose a feasibility study for the methodology.
- We prototyped the reduction of combinations.
- We prototyped the modularization of the type system.





Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Condusions

Thanks for your attention!





Software Product Lines

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

LSP +DAF

The Reductions of Combination

FOP

Saiandi Nia

Scientific Contribution

Components

Modularization

Conclusions

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





Software Product Lines

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Statem

In a Nutshell
The Reduction

The Reductions of Combinatio An Achievemen

FOF

Scientific

Contribution
Type System

Checking and Inference Modularization

Conclusions

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.





Language Product Lines

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Problem Stateme

150474

LSP &DAI

The Reduction of Combination

An Achieven

FOP

Salaadi

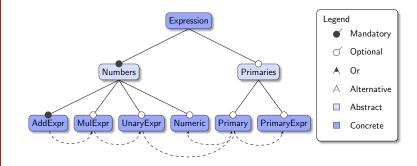
Contributio

Component

Modularizatio

Condusions

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







Language Product Lines

Universal LSP and DAP for Modular LWs

> Federico Bruzzone

Stateme

.

The Reductions of Combinatio

COE

LWs

Scientific Contribution

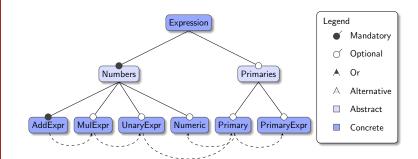
Type System

Inference

Canalysian

Condusions

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

