

Towards Self-Adaptable Languages

Gwendal Jouneaux¹ Olivier Barais¹
Benoit Combemale¹ Gunter Mussbacher²

¹Univ. Rennes, Inria, IRISA – Rennes, France

²McGill University – Montreal, Canada



UMR

IRISA



McGill

ALE Seminar — May 24, 2022



Context

Software ...



Context

Software ...

- ▶ Evolve in complex/changing environment (e.g, Cloud, embedded systems)



Context

Software ...

- ▶ Evolve in complex/changing environment (e.g, Cloud, embedded systems)
- ▶ Need dynamic adaptation to best deliver the service (e.g., Waymo¹, Netflix¹)

¹ Cf. <https://waymo.com>, <https://www.netflix.com>



Context

Software ...

- ▶ Evolve in complex/changing environment (e.g, Cloud, embedded systems)
- ▶ Need dynamic adaptation to best deliver the service (e.g., Waymo¹, Netflix¹)

Software languages ...

¹ Cf. <https://waymo.com>, <https://www.netflix.com>



Context

Software ...

- ▶ Evolve in complex/changing environment (e.g, Cloud, embedded systems)
- ▶ Need dynamic adaptation to best deliver the service (e.g., Waymo¹, Netflix¹)

Software languages ...

- ▶ Can abstract concerns into high level constructs (e.g., memory management)

¹ Cf. <https://waymo.com>, <https://www.netflix.com>

Context

Software ...

- ▶ Evolve in complex/changing environment (e.g, Cloud, embedded systems)
- ▶ Need dynamic adaptation to best deliver the service (e.g., Waymo¹, Netflix¹)

Software languages ...

- ▶ Can abstract concerns into high level constructs (e.g., memory management)

Vision : abstract self-adaption into high level language constructs

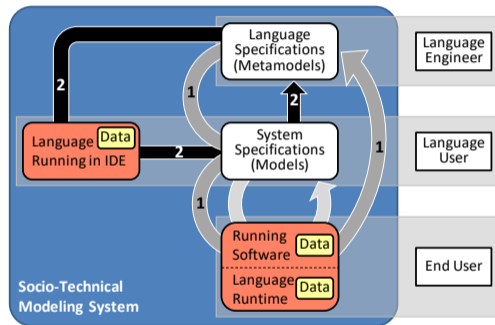
¹ Cf. <https://waymo.com>, <https://www.netflix.com>

What is a Self-Adaptable Language ?

“ A software language that abstracts the design and execution of feedback loops in the design-time environment and the run-time environment ”

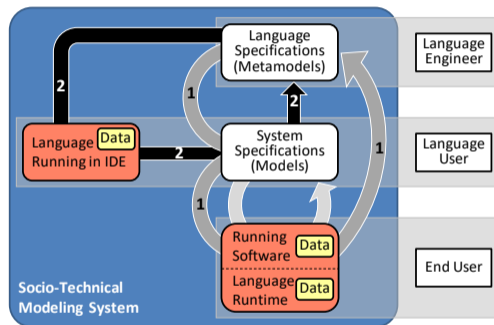
1. Free the language user from the implementation of :
 - ▶ The feedback loop
 - ▶ The trade-off analysis
2. Allow continuous and automatic evolution of itself

L-MODA | Languages, Models, and Data



L-MODA Conceptual Framework for
Self-Adaptable Languages

L-MODA | Languages, Models, and Data

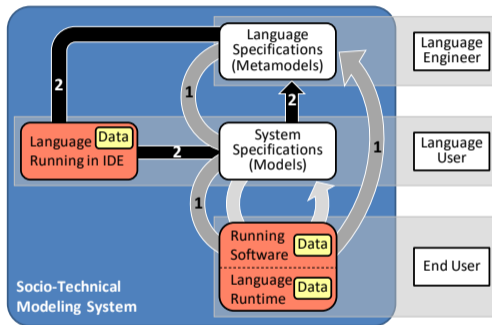


1) Runtime Feedback Loop

Use run-time data, model & metamodel
→ adaptation of language semantics

L-MODA Conceptual Framework for
Self-Adaptable Languages

L-MODA | Languages, Models, and Data



1) Runtime Feedback Loop

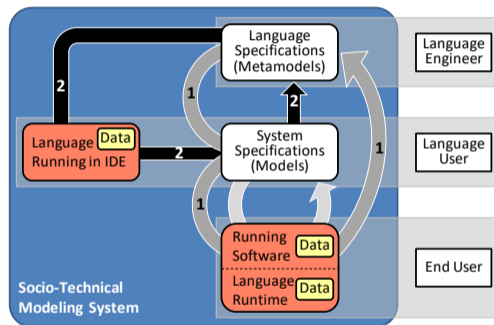
Use run-time data, model & metamodel
→ adaptation of language semantics

2) Design Feedback Loop

Use design-time data, models & metamodel
→ adaptation of syntax, pragmatics & semantics

L-MODA Conceptual Framework for
Self-Adaptable Languages

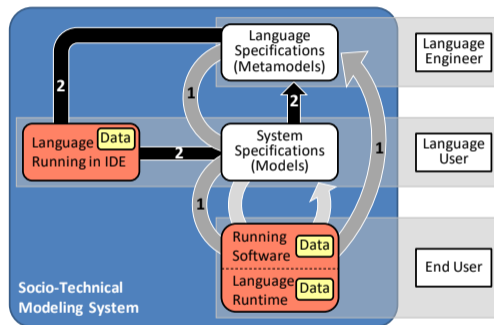
L-MODA | Stakeholders



L-MODA Conceptual Framework for
Self-Adaptable Languages

L-MODA | Stakeholders

Various uses of the feedback loops ...

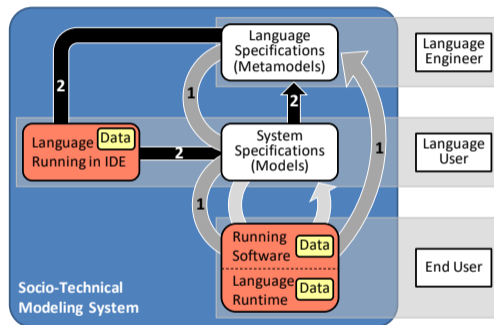


L-MODA Conceptual Framework for Self-Adaptable Languages

L-MODA | Stakeholders

Various uses of the feedback loops ...

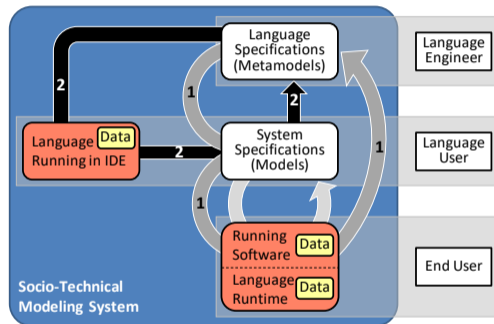
Examples for the Runtime Feedback Loop :



L-MODA Conceptual Framework for Self-Adaptable Languages

Delegation of responsibilities


L-MODA | Stakeholders



L-MODA Conceptual Framework for Self-Adaptable Languages

Various uses of the feedback loops ...

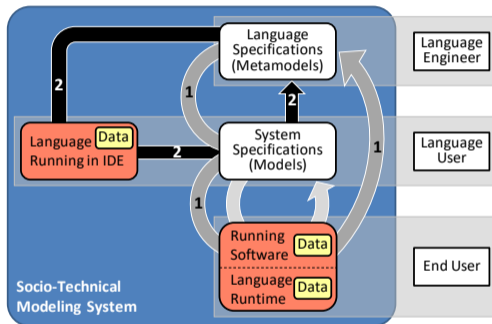
Examples for the Runtime Feedback Loop :

 Language engineer in complete control
Tailor the language to a particular trade-off

Delegation of responsibilities




L-MODA | Stakeholders




L-MODA Conceptual Framework for Self-Adaptable Languages

Various uses of the feedback loops ...

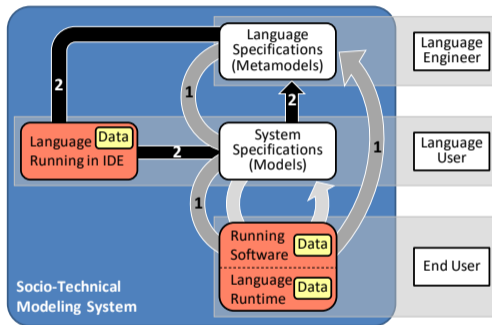
Examples for the Runtime Feedback Loop :

 **Language engineer in complete control**
Tailor the language to a particular trade-off

 **Language user custom adaptations**
Configure the adaptations for a system

Delegation of responsibilities


L-MODA | Stakeholders





L-MODA Conceptual Framework for Self-Adaptable Languages

Various uses of the feedback loops ...

Examples for the Runtime Feedback Loop :

 **Language engineer in complete control**
Tailor the language to a particular trade-off

 **Language user custom adaptations**
Configure the adaptations for a system

 **End-user preferences**
Indicate preference for trade-offs

Delegation of responsibilities



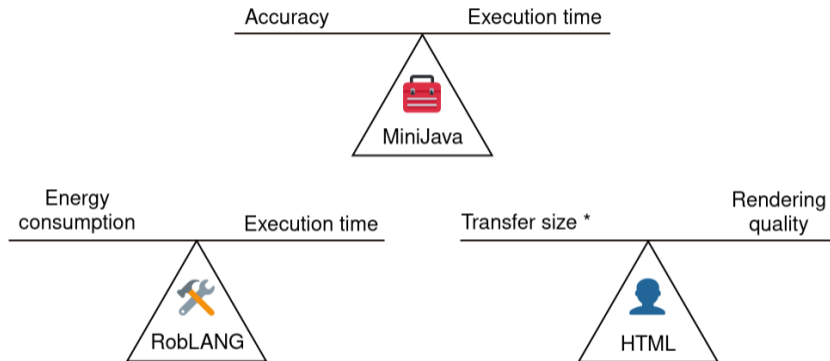
Experimentation

The case of Self-Adaptable Virtual Machines

What are Self-Adaptable Virtual Machines

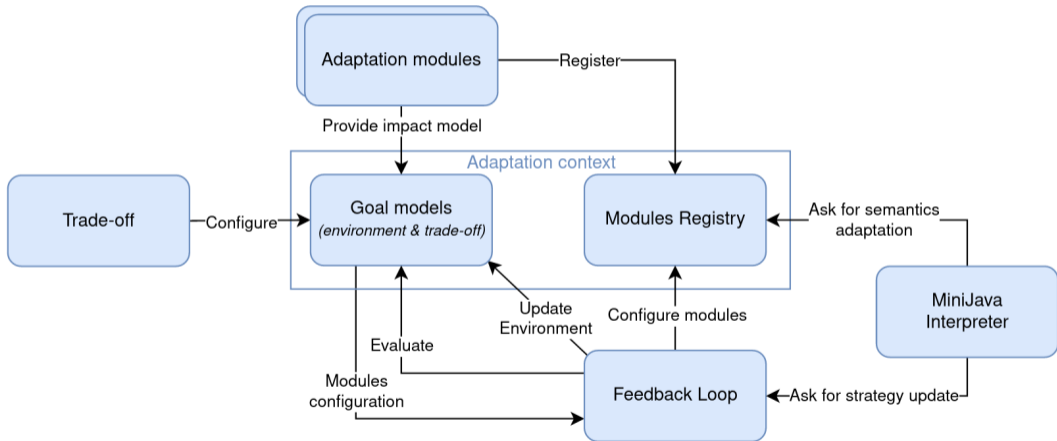
- ▶ A specific case of Self-Adaptable Languages
- ▶ Runtime Feedback loop in language operational semantics
- ▶ *In our experiment* : Pluggable architecture with delegation of responsibilities

Motivating Examples

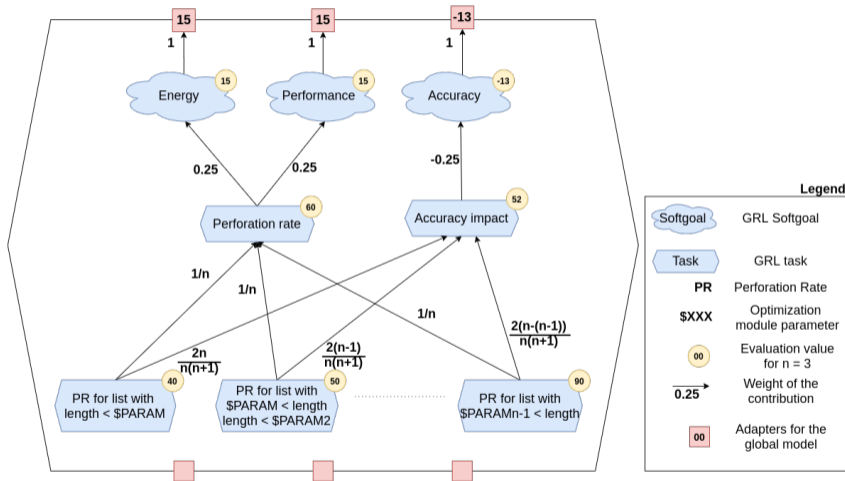


* Transfer size is proportional to energy consumption (Cf. <https://www.websitecarbon.com/>)

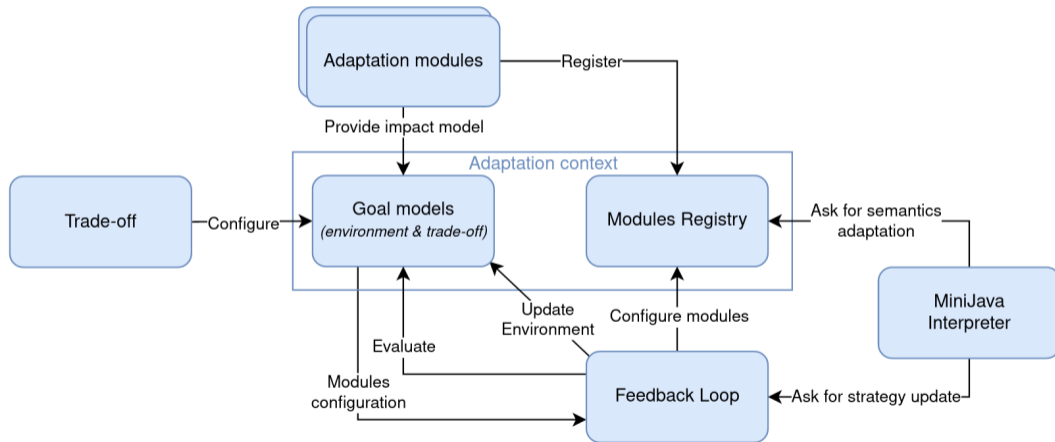
Architecture of Self-Adaptable Virtual Machines (MiniJava)



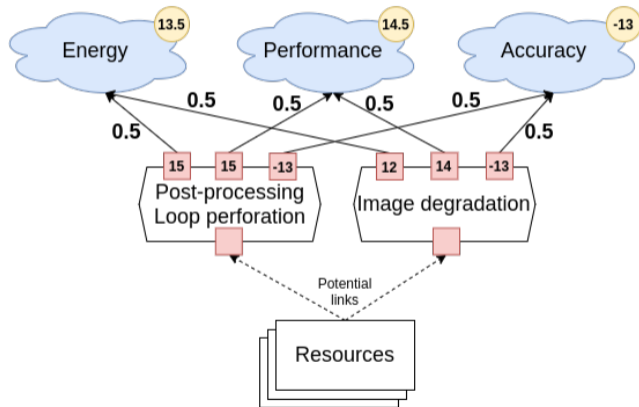
Impact Models



Architecture of Self-Adaptable Virtual Machines (MiniJava)



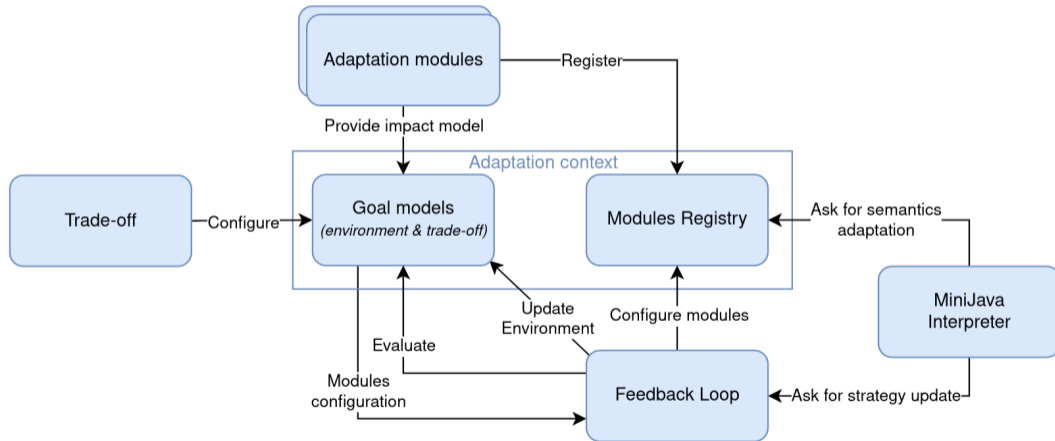
Global Goal Model



Legend

	GRL Softgoal
	Optimization module's model
	Evaluation value for $n = 3$
	Weight of the contribution
	Adapters of module's model

Architecture of Self-Adaptable Virtual Machines (MiniJava)



Adaptations

Adaptations (MiniJava)

Applied Approximate Loop Unrolling [1] on image processing algorithm (Sobel)



Standard output



Approximated output

[1] M. Rodriguez-Cancio, B. Combemale, and B. Baudry, "Approximate loop unrolling," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*, ACM, 2019

Adaptations (RobLANG)

Applied a motor speed reduction on basic actions

$$\text{Rational : } P_i = P_{max} \left(\frac{Speed_i}{Speed_{max}} \right)^3 [2]$$

Three programs studied :

- ▶ Move forward/backward
- ▶ Turn left/right
- ▶ Combination of moves and turns (square patterns)

[2] A. Al-Mofleh, S. Taib, W. Salah, *et al.*, "Importance of energy efficiency: From the perspective of electrical equipments," in *Proceedings of the 2nd International Conference on Science and Technology (ICSTIE)*, 2008

Adaptations (HTML)

amazon

compilers principles, techniques, and tools

1-16 of 315 results for "compilers principles, techniques, and tools"

Department
Books

Software Programming Compilers
Computer Hardware Design & Architecture
Compiler Design
Computer Programming
Computer Software
Mathematics
Computer Science
Operating Systems

Customer Reviews
★★★★☆ & up
★★★★☆ & up
★★★★☆ & up

Book Series
 Pragmatic Programmers
 Modern Compiler Implementation
 Use It!
 Technology in Action
 Computer science
 Make
 Essentials

Computer Platform Books
 Android
 Linux & Unix
 Macintosh
 Windows
 iPhone & iOS

Computer User Books
 Beginners & Seniors
 Advanced & Power Users
 Kids & Teens

Book Format
Paperback

Compilers: Principles, Techniques, and Tools
by Alfred Aho, Monica Lam, et al. | Aug 31, 2006
★★★★☆ - 333
Hardcover
\$79.00 to rent
Only 16 left in stock - order soon.
eTextbook
\$39.00 to rent
\$34.00 to buy
Available instantly
Paperback
\$36.00 seeeee
Ships to France
Only 1 left in stock - order soon.
More Buying Choices
\$3.06 (56 used & new offers)

Compilers: Principles, Techniques, & Tools
by Alfred V. Aho, Monica S. Lam, et al. | Jan 1, 2012
★★★★☆ - 22
Paperback

Compilers: Principles, Techniques, and Tools
by Alfred V. Aho, Ravi Sethi, et al. | Oct 1, 1985
★★★★☆ - 333
Hardcover
Paperback

- ▶ Conditional loading of resources
- ▶ Perforation of HTML lists
- ▶ Image degradation

Applied on the top 100 websites
→ 45 still deliver the content

Standard website

Adaptations (HTML)

Skip to main content

All


All Departments

compilers principles, techniques, and tools

Go

1-16 of 70 results for "compilers principles, techniques, and tools"

Sort by: Featured Sort by: Featured



Compilers: Principles, Techniques, and Tools

by Alfred Aho, Monica Lam, et al. | Aug 31, 2006

4.2 out of 5 stars (318)

Hardcover

\$79.36 \$79.36 to rent

Only 16 left in stock - order soon.

Textbook

\$39.99 \$39.99 to rent

\$74.99 to buy

Available instantly

Paperback


\$79.00 \$79.00

Ships to France

Only 2 left in stock - order soon.

More Buying Choices

\$67.78 (27 used & new offers)



- ▶ Conditional loading of resources
- ▶ Perforation of HTML lists
- ▶ Image degradation

Applied on the top 100 websites
→ 45 still deliver the content

Adapted website

Evaluate the relevance of proposed adaptation

TL;DR : Good results but ...

- ▶ Correct adaptations of MiniJava
- ▶ Performance overhead
- ▶ Up to 10x more actions on RobLANG
- ▶ Lack of control on the adaptations
- ▶ Energy reduction from -8.7% to 97.2% with a mean of 63.8% [54.2%, 73.4%]
- ▶ Deal with the diversity of programs oblivious of the adaptations performed

Compare Language-level vs System-level abstractions

Attempt to implement as library of the language

VMs	Feedback loop	Trade-off reasoning	Feedback loop calls	Interaction with the domain
MiniJava	=	=	+	+
RobLang	=	+ +	+	-
HTML (JS)	=	=	=	-

Comparison of implementation simplicity (+ in favor of language-level)

Conclusion

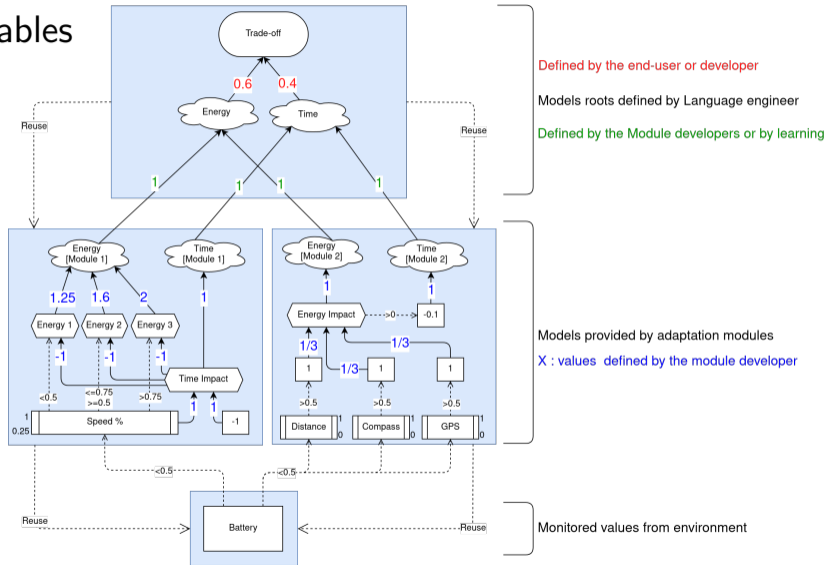
- ▶ The concept of Self-Adaptable Language and its conceptual framework
- ▶ A framework to implement Self-Adaptable Virtual Machines
- ▶ Ongoing work on specification of adaptive semantics

Perspectives



- ▶ DSLs to specify adaptations and impacts
- ▶ Formalisation of adaptive operational semantics
- ▶ Testing and debugging programs defined using SALs

Thanks for your attention !

Goal Model with Variables



References

-  M. Rodriguez-Cancio, B. Combemale, and B. Baudry, "Approximate loop unrolling," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*, ACM, 2019.
-  A. Al-Mofleh, S. Taib, W. Salah, and M. Azizan, "Importance of energy efficiency: From the perspective of electrical equipments," in *Proceedings of the 2nd International Conference on Science and Technology (ICSTIE)*, 2008.