Towards Self-Adaptable Languages

Gwendal Jouneaux [1] Olivier Barais [1]
Benoit Combemale [1] Gunter Mussbacher [2]

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

[2]McGill University – Montreal, Canada

Onward! — October 20, 2021

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., Waymo[1], Netflix[1])

[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., Waymo[1], Netflix[1])

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., Waymo[1], Netflix[1])

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., Waymo[1], Netflix[1])

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`

Contribution overview

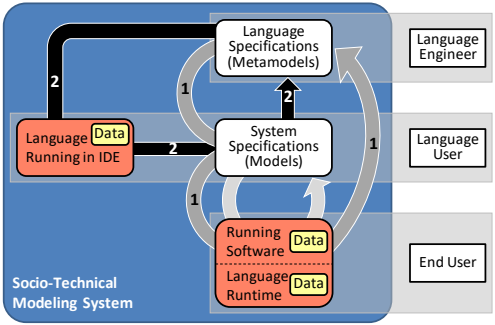
1. The novel concept of Self-Adaptable Language
2. A conceptual reference framework (L-MODA)
3. Emerging results on the concept of Self-Adaptable Virtual Machines
4. A research road-map for Self-Adaptable Languages

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* "

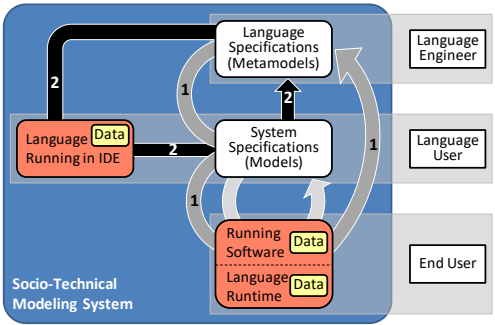
1. Free the language user from the implementation of :
   - The feedback loop
   - The trade-off analysis
2. Allow continuous and automatic evolution of itself

L-MODA | Languages, Models, and Data



L-MODA Reference Framework for Self-Adaptable Languages

L-MODA | Languages, Models, and Data

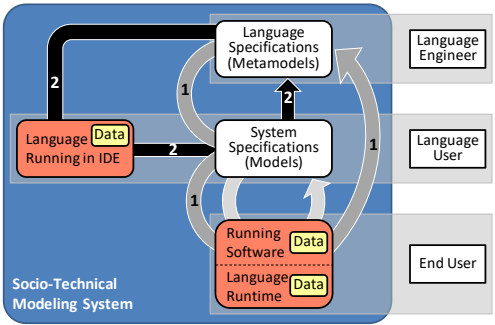


L-MODA Reference Framework for Self-Adaptable Languages

1) Runtime Feedback Loop

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

L-MODA | Languages, Models, and Data



L-MODA Reference Framework for Self-Adaptable Languages

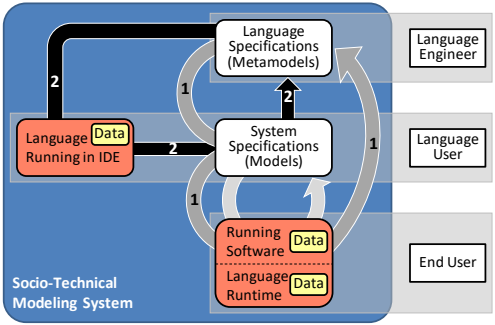
1) Runtime Feedback Loop

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

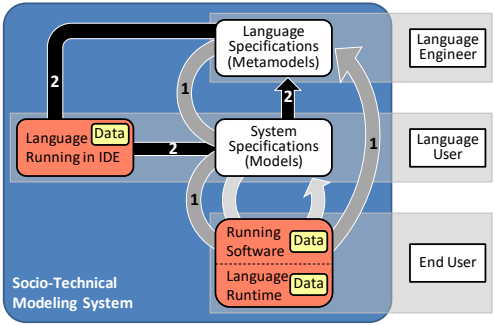
L-MODA | Stakeholders



L-MODA Reference Framework for Self-Adaptable Languages

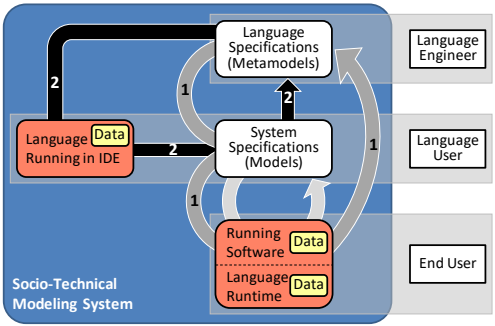
L-MODA | Stakeholders

Various uses of the feedback loops ...



L-MODA Reference Framework for Self-Adaptable Languages

L-MODA | Stakeholders



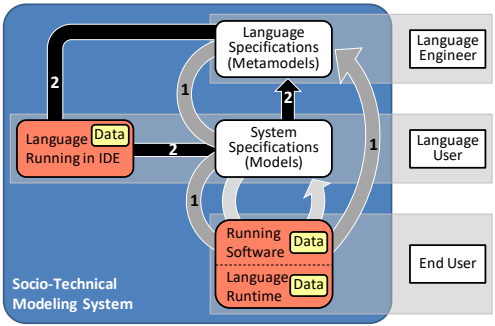
L-MODA Reference Framework for Self-Adaptable Languages

Various uses of the feedback loops ...

Examples for the Runtime Feedback Loop :

Delegation of responsabilities

L-MODA | Stakeholders



L-MODA Reference Framework for Self-Adaptable Languages

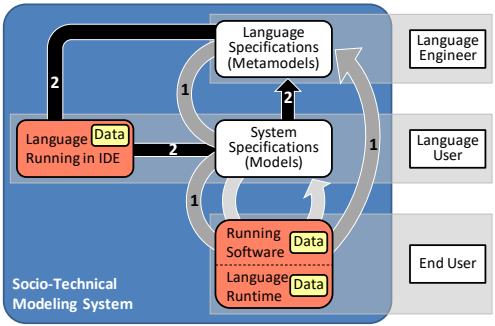
Various uses of the feedback loops ...

Examples for the Runtime Feedback Loop :

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

Delegation of responsabilities

L-MODA | Stakeholders



L-MODA Reference Framework for Self-Adaptable Languages

Various uses of the feedback loops ...

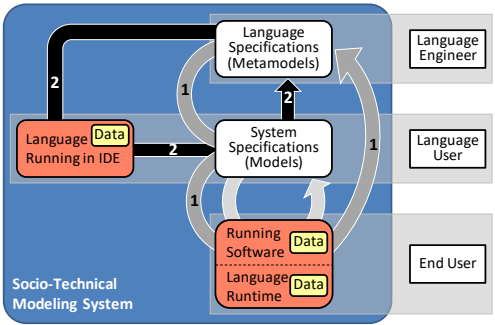
Examples for the Runtime Feedback Loop :

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 responsabilities

L-MODA | Stakeholders



L-MODA Reference 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 responsabilities

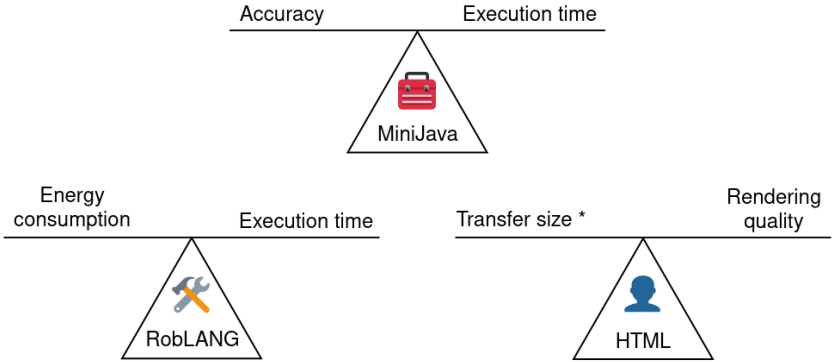
Experimentation

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

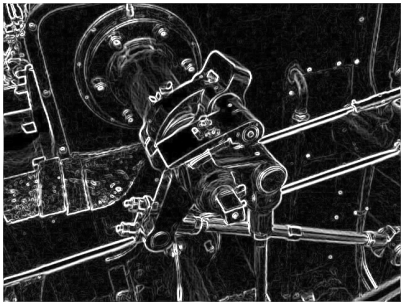


* Transfer size is proportional to energy consumption (Cf. `https://www.websitecarbon.com/`)

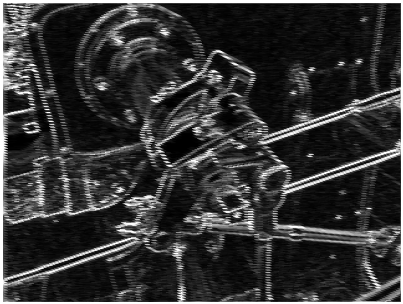
Adaptations

Adaptations (MiniJava)

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



Standard output

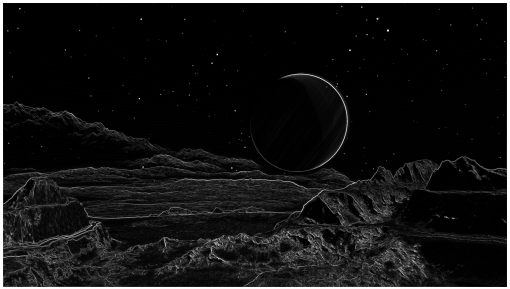


Approximated output

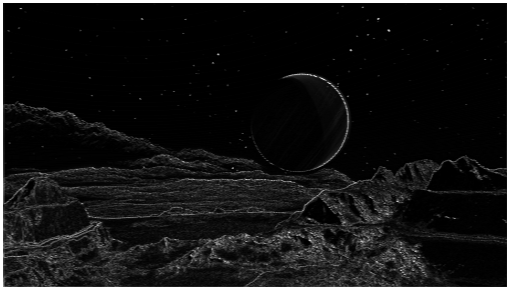
[1] M. Rodriguez-Cancio, B. Combemale, and B. Baudry, "Approximate loop unrolling," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*, ACM, 2019

Adaptations (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

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

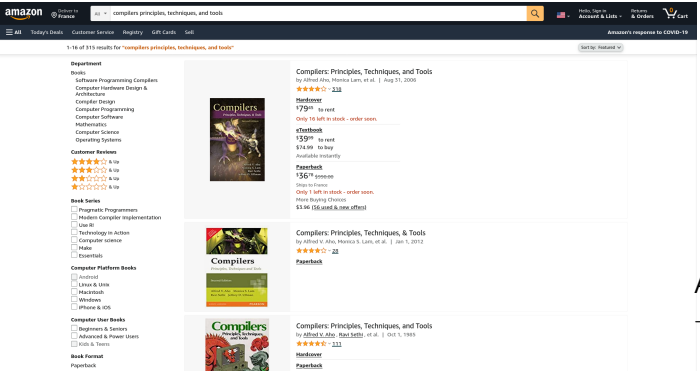
Three programs studied :

- 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)

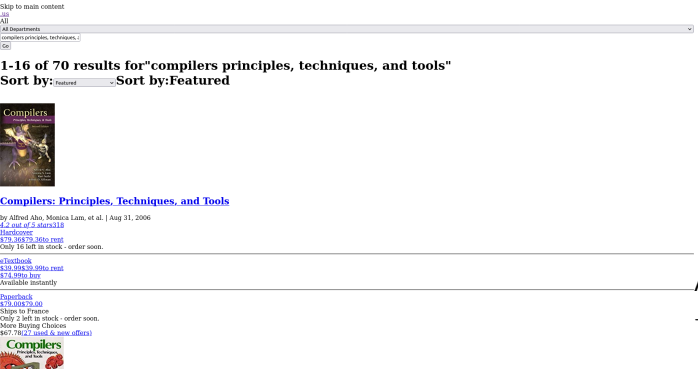


Standard website

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

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

Adaptations (HTML)



Adapted website

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

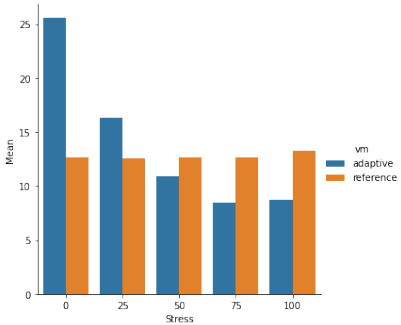
Applied on the top 100 websites
$\rightarrow$ 45 still deliver the content

Relevance of proposed adaptation

TL;DR : Good results but ...

- Performance overhead
- Deal with the diversity of programs oblivious of the adaptations performed
- Lack of control on the adaptations

Relevance of proposed adaptation (MiniJava)



Sobel filter mean execution time depending on the CPU usage

- Correct adaptation behavior
- Results emphasize the need for a more efficient implementation of the adaptation system

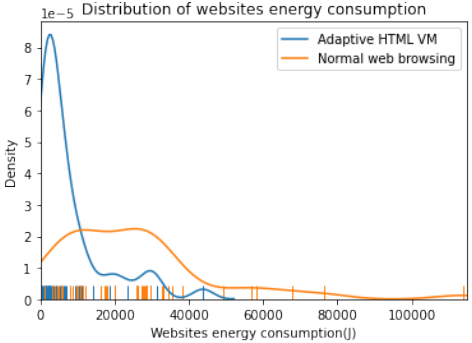
Relevance of proposed adaptation (RobLANG)

Prog.	Relaxation of the time constraint (in %)			
	0% *	25%	50%	75%
Turn	288	524 (182%)	1164 (404%)	3018 (1048%)
Move	87	154 (177%)	330 (379%)	860 (989%)
Square	74	142 (192%)	294 (397%)	794 (1073%)
Mean	100%	183.56%	393.19%	1036.06%

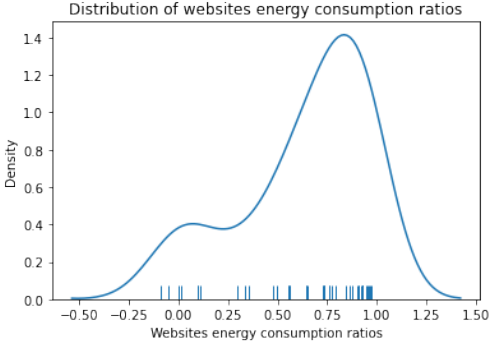
(*) We use the original interpreter as baseline.

Number of actions performed by robot (and relative enhancements compared to the baseline) depending on the program and the percentage of time constraint relaxation.

Relevance of proposed adaptation (HTML)



Websites consumption when browsing normally versus using the HTML SAVM



Distribution of energy reduction ratios

Language-level vs System-level abstraction

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
- Proof-of-concept implementation of Self-Adaptable Virtual Machines
- Promising results for adaptations of language operational semantics

Future work

Support of the Runtime Feedback Loop

- Feedback loop configuration
- A reference framework for common implementation [3]
- ...

Support of the Design Feedback Loop

- Model the development context
- Detect evolution opportunities
- Navigate in evolution of programs
- ...

[3] G. Jouneaux, O. Barais, B. Combemale, *et al.*, "SEALS: A framework for building Self-Adaptive Virtual Machines," in *Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering (SLE '21)*, Chicago, United States, Oct. 2021. DOI: 10.1145/3486608.3486912

Future work

Support of the Runtime Feedback Loop

- Feedback loop configuration
- A reference framework for common implementation [3]
- ...

Support of the Design Feedback Loop

- Model the development context
- Detect evolution opportunities
- Navigate in evolution of programs
- ...

For more details take a look at our research roadmap

[3] G. Jouneaux, O. Barais, B. Combemale, *et al.*, "SEALS: A framework for building Self-Adaptive Virtual Machines," in *Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering (SLE '21)*, Chicago, United States, Oct. 2021. DOI: 10.1145/3486608.3486912

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.

G. Jouneaux, O. Barais, B. Combemale, and G. Mussbacher, "SEALS: A framework for building Self-Adaptive Virtual Machines," in *Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering (SLE '21)*, Chicago, United States, Oct. 2021. DOI: 10.1145/3486608.3486912.