Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design

dr. Vadim Zaytsev, University of Twente

MLE@MoDELS 2024, 22 September 2024

UNIVERSITY OF TWENTE.

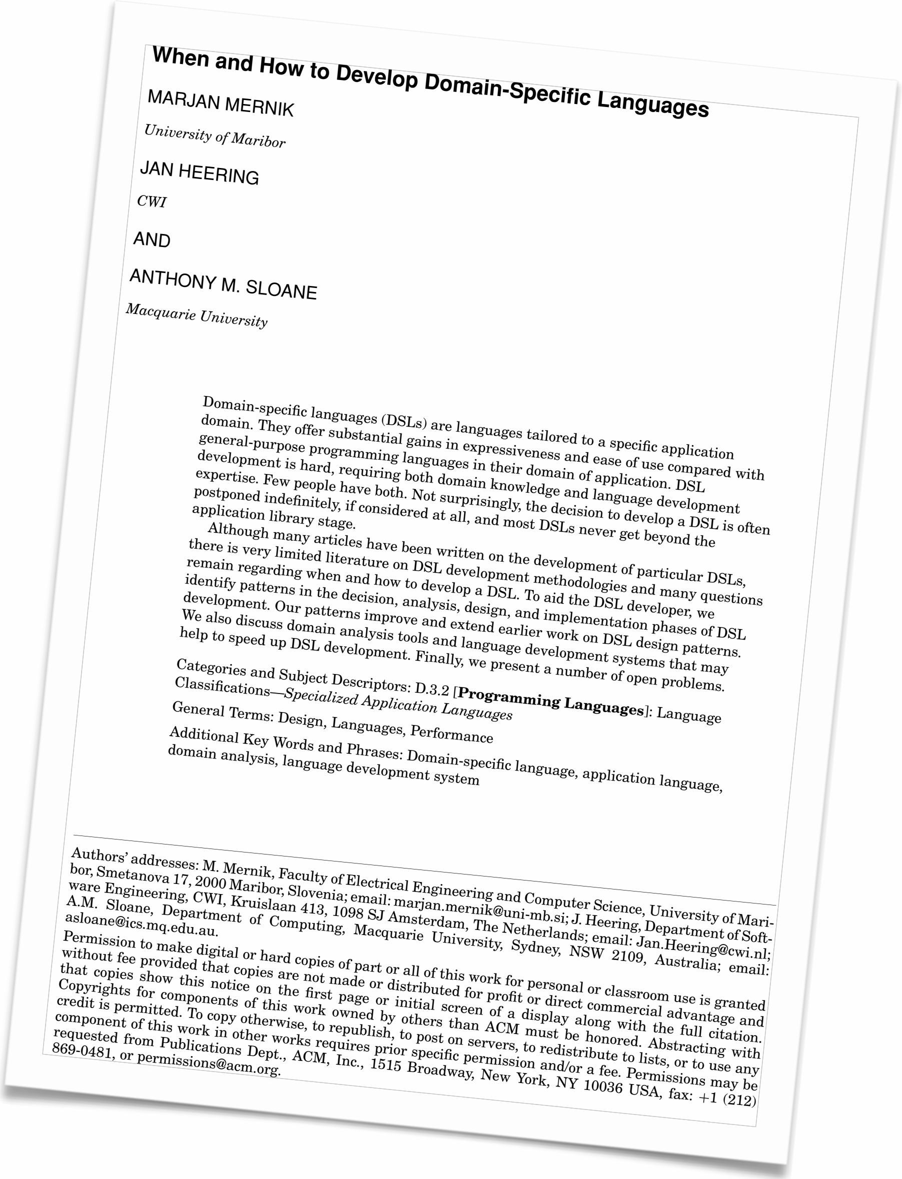
Zorro

Zero downtime in cyber physical systems

UNIVERSITY OF TWENTE.

Formal Methods & Tools

When and How to Develop DSLs



When and How to Develop Domain-Specific Languages

MARJAN MERNIK

University of Maribor

JAN HEERING

CWI

AND

ANTHONY M. SLOANE

Macquarie University

Domain-specific languages (DSLs) are languages tailored to a specific application domain. They offer substantial gains in expressiveness and ease of use compared with general-purpose programming languages in their domain of application. DSL development is hard, requiring both domain knowledge and language development expertise. Few people have both. Not surprisingly, the decision to develop a DSL is often postponed indefinitely, if considered at all, and most DSLs never get beyond the application library stage.

Although many articles have been written on the development of particular DSLs, there is very limited literature on DSL development methodologies and many questions remain regarding when and how to develop a DSL. To aid the DSL developer, we identify patterns in the decision, analysis, design, and implementation phases of DSL development. Our patterns improve and extend earlier work on DSL design patterns. We also discuss domain analysis tools and language development systems that may help to speed up DSL development. Finally, we present a number of open problems.

Categories and Subject Descriptors: D.3.2 [**Programming Languages**]: Language Classifications—*Specialized Application Languages*

General Terms: Design, Languages, Performance

Additional Key Words and Phrases: Domain-specific language, application language, domain analysis, language development system

Authors' addresses: M. Mernik, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: marjan.mernik@uni-mb.si; J. Heering, Department of Software Engineering, CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands; email: Jan.Heering@cwi.nl; A.M. Sloane, Department of Computing, Macquarie University, Sydney, NSW 2109, Australia; email: asloane@ics.mq.edu.au.

DOI: 10.1145/1118890.1118892

Modelling of CPS in DSL Decision, Analysis, Design

Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design

Marcus Gerhold
m.gerhold@utwente.nl
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Aliaksei Kouzel
a.kouzel@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Haroun Mangal
h.mangal@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Selin A. Mehmed
s.a.mehmed@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Vadim Zaytsev
vadim@grammarware.net
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Abstract

Cyber-Physical Systems (CPS) integrate computational algorithms and physical components, requiring sophisticated modelling techniques to address complex interactions and dynamics. This paper explores the creation of Domain-Specific Languages (DSLs) tailored for CPS, focusing on the initial three critical phases: decision, analysis, design. We present four key aspects to address in the decision phase, design an ontology as a domain model for the analysis phase, and collect some advice for the design phase. By systematically addressing these phases, we provide a comprehensive framework for developing DSLs that can efficiently model CPS, facilitating improved design, verification, and deployment of these intricate systems.

CCS Concepts

• Software and its engineering → Domain specific languages; Interoperability; Design languages; • Information systems → Ontologies; • Networks → Network protocol design.

Keywords

Cyber-Physical Systems, Ontological Analysis, Domain-Specific Languages

ACM Reference Format:
Marcus Gerhold, Aliaksei Kouzel, Haroun Mangal, Selin A. Mehmed, and Vadim Zaytsev. 2024. Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688348

1 Introduction

Cyber-Physical Systems (CPS) represent collaborations of computational algorithms and physical components [61], creating a network where digital systems monitor and control physical processes through sensors and actuators. These systems form a feedback loop where sensor data informs computational decisions, and actuators execute these decisions to affect the physical environment. Examples of CPS include smart grids, autonomous vehicles, and advanced medical monitoring systems [66, 83]. CPS hold substantial potential across diverse domains such as smart manufacturing, robotics, healthcare, intelligent transportation, and smart cities, offering benefits like enhanced automation, improved safety, and optimised resource usage [35, 42, 61, 72].

Realising the full potential of CPS poses significant challenges due to their inherent complexity and heterogeneity [83]. Integrating the continuous, concurrent physical world with the discrete, sequential cyber world often leads to non-deterministic behaviours, complicating the development of reliable and dependable models. Model-Driven Engineering (MDE) addresses these complexities by breaking down CPS into manageable components, yet the intricacies of such systems demand specialised modelling approaches [22].

Domain-Specific Languages (DSLs) offer a promising solution for CPS modelling by providing tailored notations and constructs specific to the domain [23, 87]. Unlike general-purpose languages (GPLs), DSLs can encapsulate domain-specific knowledge, simplifying the modelling process and enhancing communication between domain experts and developers [59]. For instance, UML and VHDL are well-known DSLs in their respective domains.

It is known from prior work that DSL development can be conceptually split into phases of **decision**, **analysis**, **design** and **implementation** [60]. By addressing these phases, we wish to provide a comprehensive framework for developing DSLs that can efficiently model CPS. Essentially we want the answers to the following research questions:

RQ1: What are critical aspects that DSLs must address to ensure that modelling and implementation of CPS to be most effective?
RQ2: How can we analyse and model the CPS domain to understand the foundational concepts and relationships that underpin these systems?

DOI: 10.1145/3652620.3688348

When and How: Phases

Decision

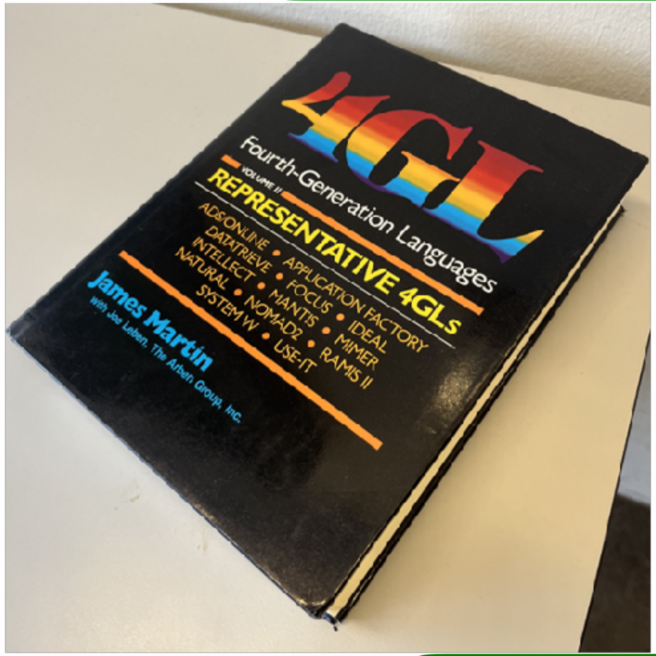
Analysis

Design

Implementation

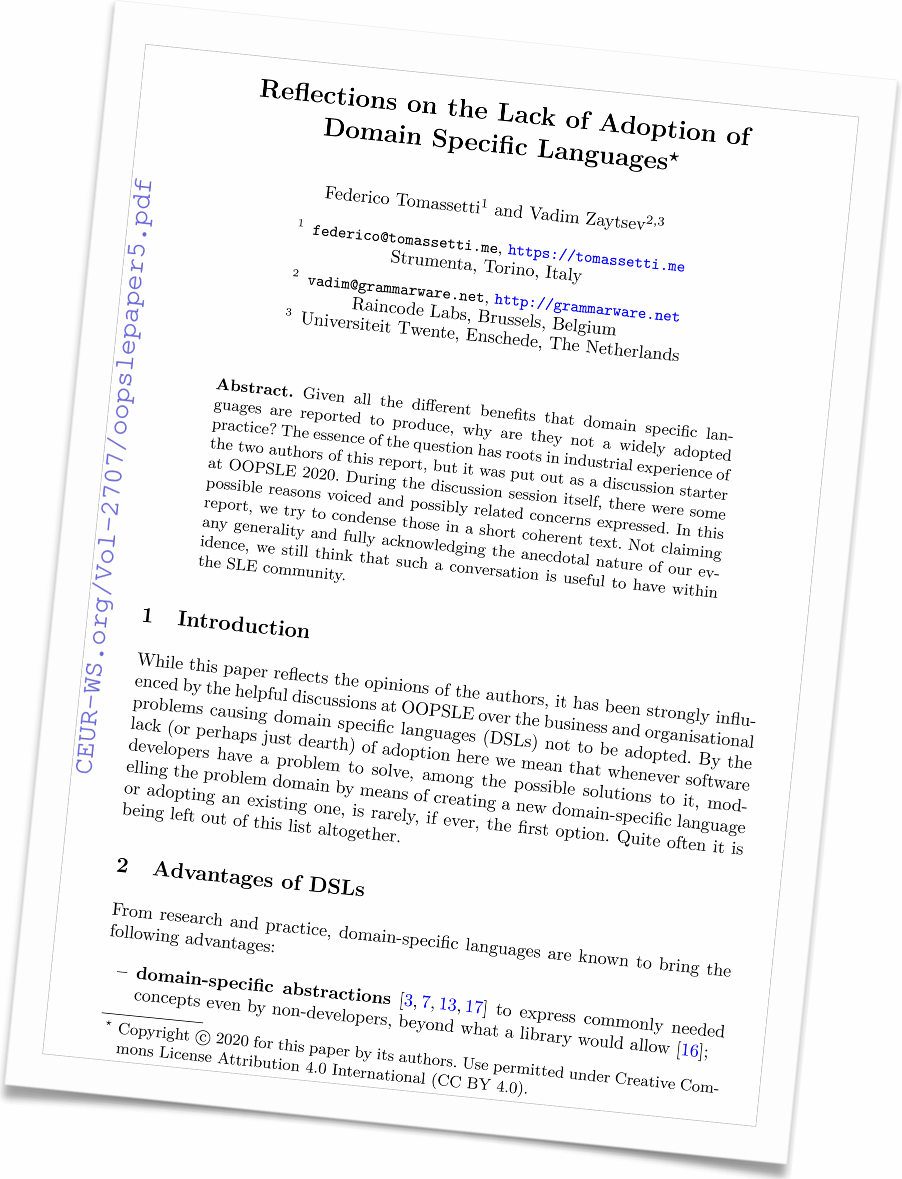
Deployment

Retirement



Why DSL?

- domain-specific abstractions
- domain-specific notations
- separation of concerns
- tool support
- conciseness/self-documentation
- productivity/maintainability
- reliability & …ilities
- conservation/reuse of domain knowledge
- executability/liveness
- involvement/collaboration
- shorter lifespan



CEUR-WS.org/Vol-2707/oopslepaper5.pdf

Reflections on the Lack of Adoption of Domain Specific Languages*

Federico Tomassetti[1] and Vadim Zaytsev[2,3]

[1] federico@tomassetti.me, https://tomassetti.me
Strumenta, Torino, Italy
[2] vadim@grammarware.net, http://grammarware.net
Raincode Labs, Brussels, Belgium
[3] Universiteit Twente, Enschede, The Netherlands

Abstract. Given all the different benefits that domain specific languages are reported to produce, why are they not a widely adopted practice? The essence of the question has roots in industrial experience of the two authors of this report, but it was put out as a discussion starter at OOPSLE 2020. During the discussion session itself, there were some possible reasons voiced and possibly related concerns expressed. In this report, we try to condense those in a short coherent text. Not claiming any generality and fully acknowledging the anecdotal nature of our evidence, we still think that such a conversation is useful to have within the SLE community.

1 Introduction

While this paper reflects the opinions of the authors, it has been strongly influenced by the helpful discussions at OOPSLE over the business and organisational problems causing domain specific languages (DSLs) not to be adopted. By the lack (or perhaps just dearth) of adoption here we mean that whenever software developers have a problem to solve, among the possible solutions to it, modelling the problem domain by means of creating a new domain-specific language or adopting an existing one, is rarely, if ever, the first option. Quite often it is being left out of this list altogether.

2 Advantages of DSLs

From research and practice, domain-specific languages are known to bring the following advantages:

– **domain-specific abstractions** [3, 7, 13, 17] to express commonly needed concepts even by non-developers, beyond what a library would allow [16];

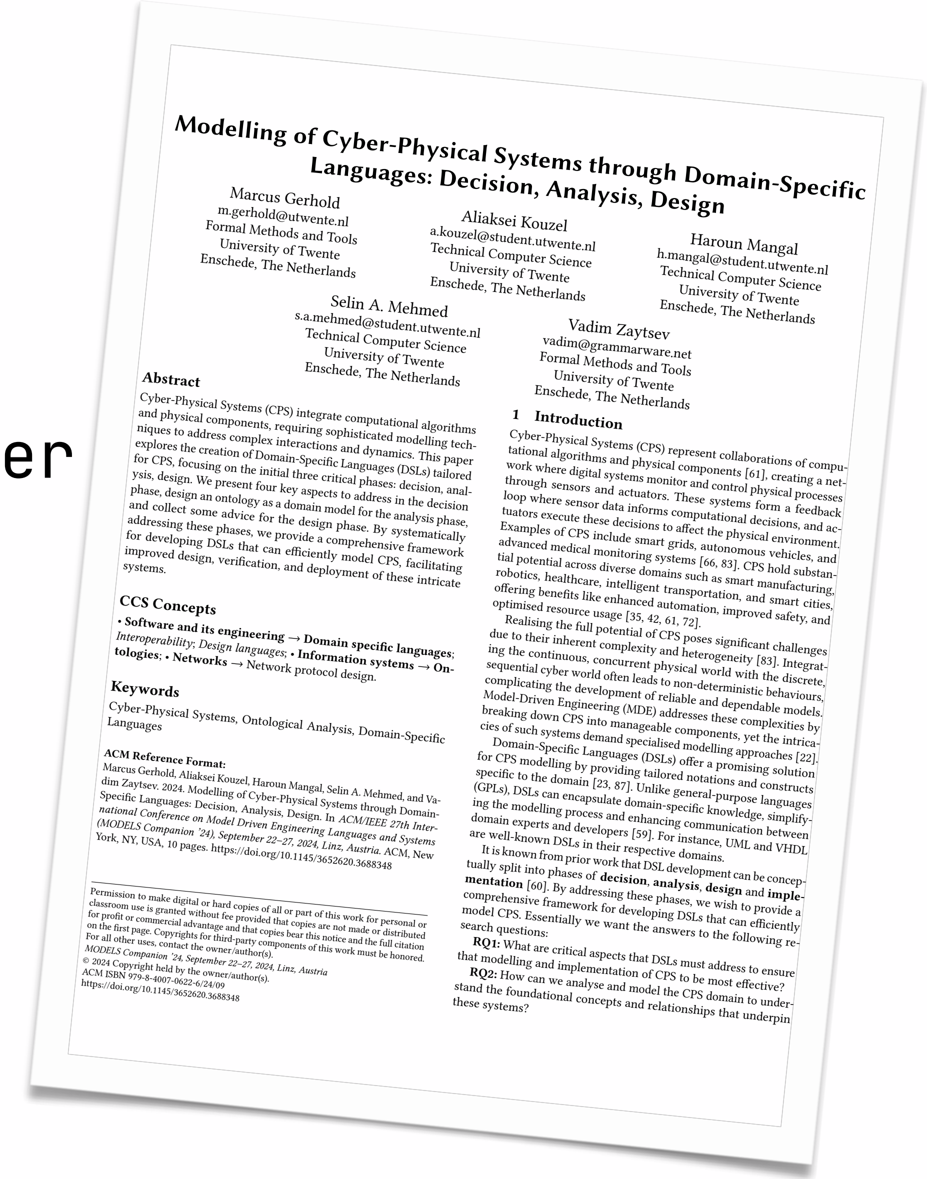
CEUR: 2707.5

Decision Phase

- **domain**
  - catering, lithography, music, grammars
- **codomain**
  - ecosystem, platform, integration
- **paradigm**
  - declarative, interactive, traversal-driven
- **purpose**
  - automation, visualisation
- **approach**
  - visual, textual, API, library

Decision

- **interoperability**
  - specify how components work together
- **behaviour**
  - define the flow of operation
- **timing**
  - delays, deadlines, sheduling
- **discrete&continuous**
  - physical process vs computation



Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design

Marcus Gerhold
m.gerhold@utwente.nl
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Aliaksei Kouzel
a.kouzel@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Haroun Mangal
h.mangal@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Selin A. Mehmed
s.a.mehmed@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Vadim Zaytsev
vadim@grammarware.net
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Abstract

Cyber-Physical Systems (CPS) integrate computational algorithms and physical components, requiring sophisticated modelling techniques to address complex interactions and dynamics. This paper explores the creation of Domain-Specific Languages (DSLs) tailored for CPS, focusing on the initial three critical phases: decision, analysis, design. We present four key aspects to address in the decision phase, design an ontology as a domain model for the analysis phase, and collect some advice for the design phase. By systematically addressing these phases, we provide a comprehensive framework for developing DSLs that can efficiently model CPS, facilitating improved design, verification, and deployment of these intricate systems.

CCS Concepts

• **Software and its engineering** → **Domain specific languages**; *Interoperability*; *Design languages*; • **Information systems** → **Ontologies**; • **Networks** → Network protocol design.

Keywords

Cyber-Physical Systems, Ontological Analysis, Domain-Specific Languages

ACM Reference Format:
Marcus Gerhold, Aliaksei Kouzel, Haroun Mangal, Selin A. Mehmed, and Vadim Zaytsev. 2024. Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688348

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3688348

1 Introduction

Cyber-Physical Systems (CPS) represent collaborations of computational algorithms and physical components [61], creating a network where digital systems monitor and control physical processes through sensors and actuators. These systems form a feedback loop where sensor data informs computational decisions, and actuators execute these decisions to affect the physical environment. Examples of CPS include smart grids, autonomous vehicles, and advanced medical monitoring systems [66, 83]. CPS hold substantial potential across diverse domains such as smart manufacturing, robotics, healthcare, intelligent transportation, and smart cities, offering benefits like enhanced automation, improved safety, and optimised resource usage [35, 42, 61, 72].

Realising the full potential of CPS poses significant challenges due to their inherent complexity and heterogeneity [83]. Integrating the continuous, concurrent physical world with the discrete, sequential cyber world often leads to non-deterministic behaviours, complicating the development of reliable and dependable models. Model-Driven Engineering (MDE) addresses these complexities by breaking down CPS into manageable components, yet the intricacies of such systems demand specialised modelling approaches [22].

Domain-Specific Languages (DSLs) offer a promising solution for CPS modelling by providing tailored notations and constructs specific to the domain [23, 87]. Unlike general-purpose languages (GPLs), DSLs can encapsulate domain-specific knowledge, simplifying the modelling process and enhancing communication between domain experts and developers [59]. For instance, UML and VHDL are well-known DSLs in their respective domains.

It is known from prior work that DSL development can be conceptually split into phases of **decision**, **analysis**, **design** and **implementation** [60]. By addressing these phases, we wish to provide a comprehensive framework for developing DSLs that can efficiently model CPS. Essentially we want the answers to the following research questions:

RQ1: What are critical aspects that DSLs must address to ensure that modelling and implementation of CPS to be most effective?

RQ2: How can we analyse and model the CPS domain to understand the foundational concepts and relationships that underpin these systems?

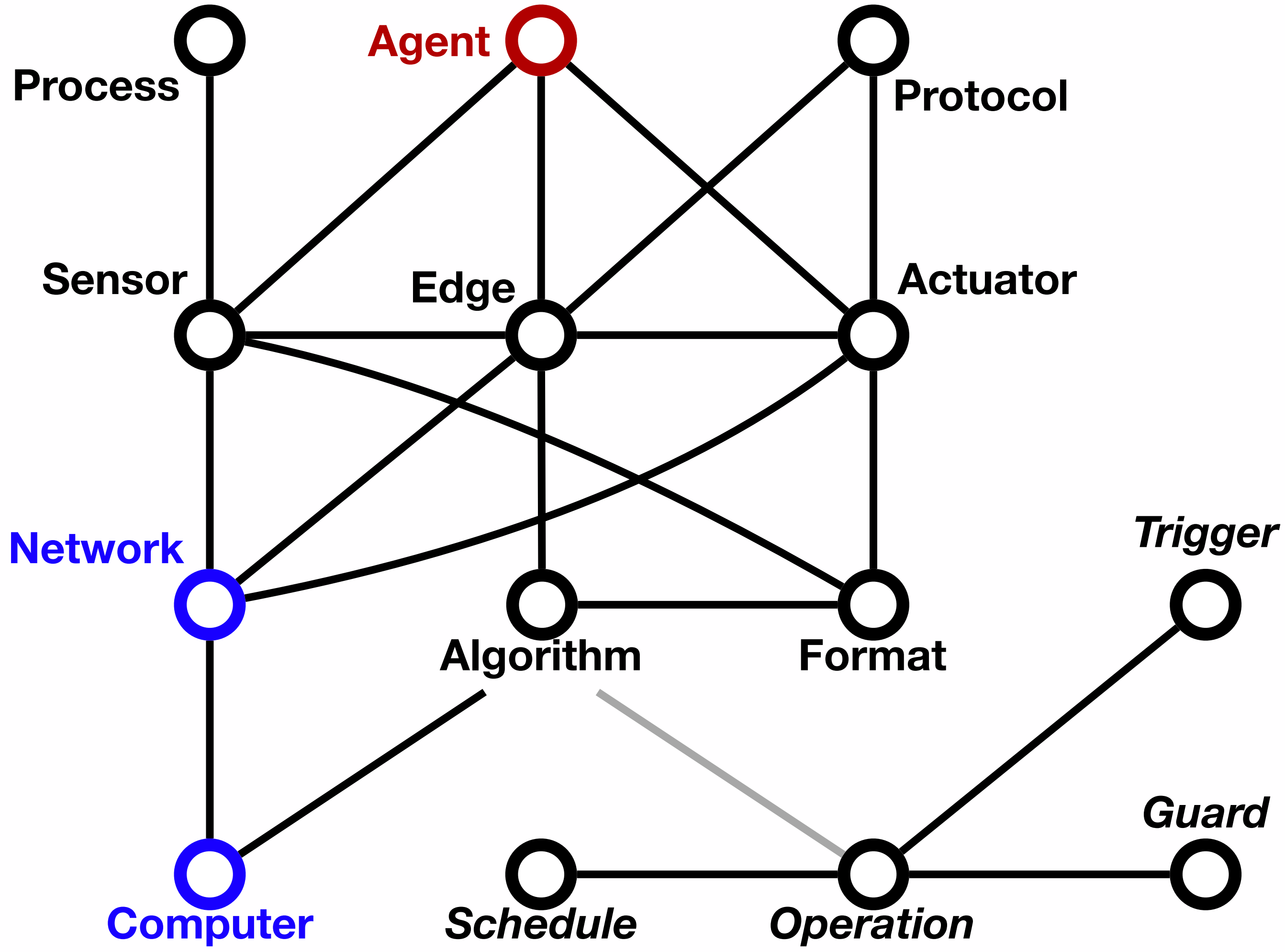
DOI: 10.1145/3652620.3688348

UNIVERSITY OF TWENTE. | Formal Methods & Tools

Analysis Phase

- terminology
  - concepts, aspects, definitions
- knowledge
  - capture, representation, management
- methodology
  - DARE, DSSA, FAST, FODA, ODE, ODM, …
- models
  - ontology, feature model
- formal vs informal
  - text, diagrams, algebra, logic

Analysis



Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design

Marcus Gerhold
m.gerhold@utwente.nl
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Aliaksei Kouzel
a.kouzel@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Haroun Mangal
h.mangal@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Selin A. Mehmed
s.a.mehmed@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Vadim Zaytsev
vadim@grammarware.net
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Abstract

Cyber-Physical Systems (CPS) integrate computational algorithms and physical components, requiring sophisticated modelling techniques to address complex interactions and dynamics. This paper explores the creation of Domain-Specific Languages (DSLs) tailored for CPS, focusing on the initial three critical phases: decision, analysis, design. We present four key aspects to address in the decision phase, design an ontology as a domain model for the analysis phase, and collect some advice for the design phase. By systematically addressing these phases, we provide a comprehensive framework for developing DSLs that can efficiently model CPS, facilitating improved design, verification, and deployment of these intricate systems.

CCS Concepts

• **Software and its engineering** → **Domain specific languages**; *Interoperability; Design languages*; • **Information systems** → **Ontologies**; • **Networks** → Network protocol design.

Keywords

Cyber-Physical Systems, Ontological Analysis, Domain-Specific Languages

ACM Reference Format:

Marcus Gerhold, Aliaksei Kouzel, Haroun Mangal, Selin A. Mehmed, and Vadim Zaytsev. 2024. Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688348

1 Introduction

Cyber-Physical Systems (CPS) represent collaborations of computational algorithms and physical components [61], creating a network where digital systems monitor and control physical processes through sensors and actuators. These systems form a feedback loop where sensor data informs computational decisions, and actuators execute these decisions to affect the physical environment. Examples of CPS include smart grids, autonomous vehicles, and advanced medical monitoring systems [66, 83]. CPS hold substantial potential across diverse domains such as smart manufacturing, robotics, healthcare, intelligent transportation, and smart cities, offering benefits like enhanced automation, improved safety, and optimised resource usage [35, 42, 61, 72].

Realising the full potential of CPS poses significant challenges due to their inherent complexity and heterogeneity [83]. Integrating the continuous, concurrent physical world with the discrete, sequential cyber world often leads to non-deterministic behaviours, complicating the development of reliable and dependable models. Model-Driven Engineering (MDE) addresses these complexities by breaking down CPS into manageable components, yet the intricacies of such systems demand specialised modelling approaches [22].

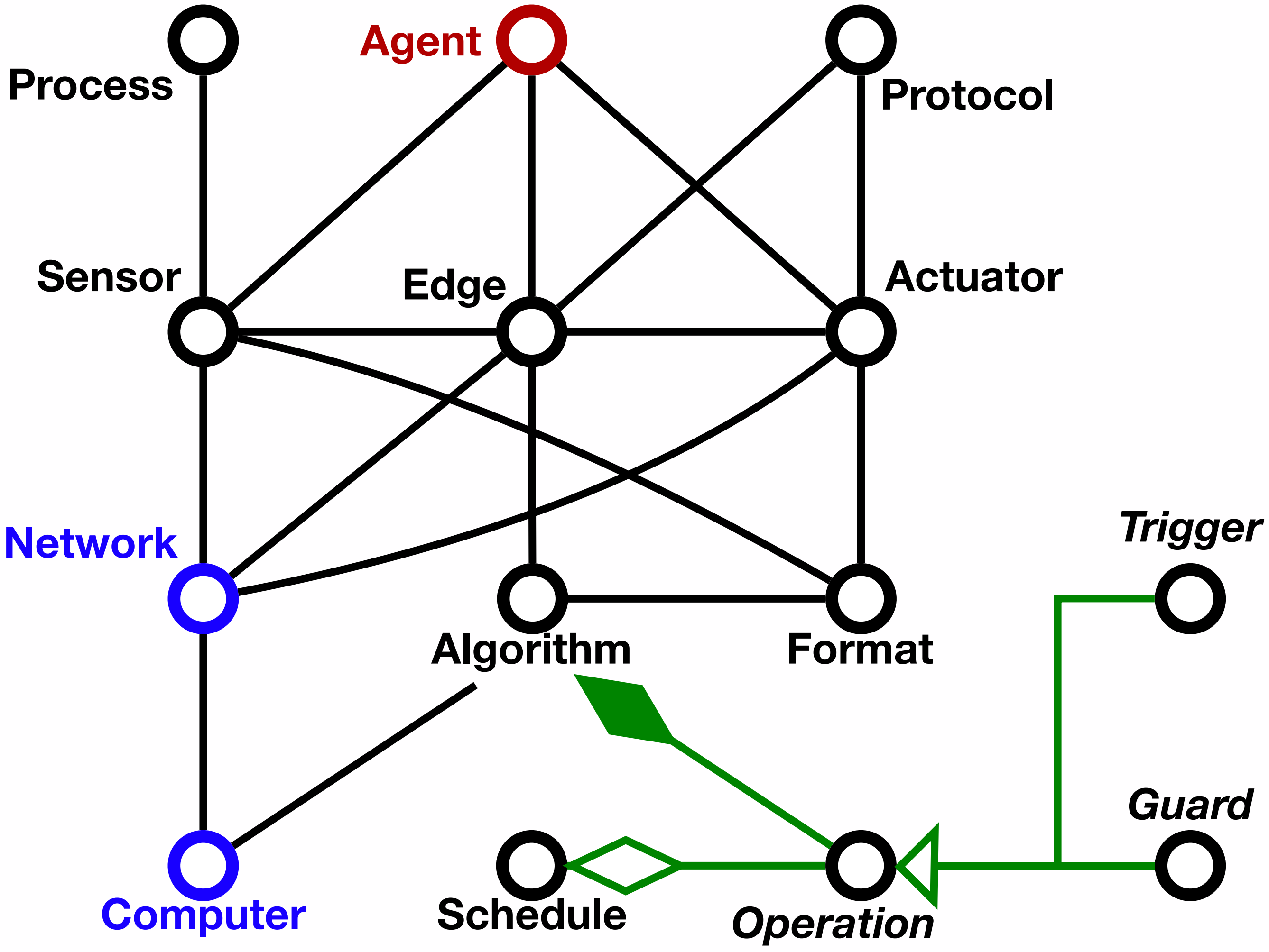
Domain-Specific Languages (DSLs) offer a promising solution for CPS modelling by providing tailored notations and constructs specific to the domain [23, 87]. Unlike general-purpose languages (GPLs), DSLs can encapsulate domain-specific knowledge, simplifying the modelling process and enhancing communication between domain experts and developers [59]. For instance, UML and VHDL are well-known DSLs in their respective domains.

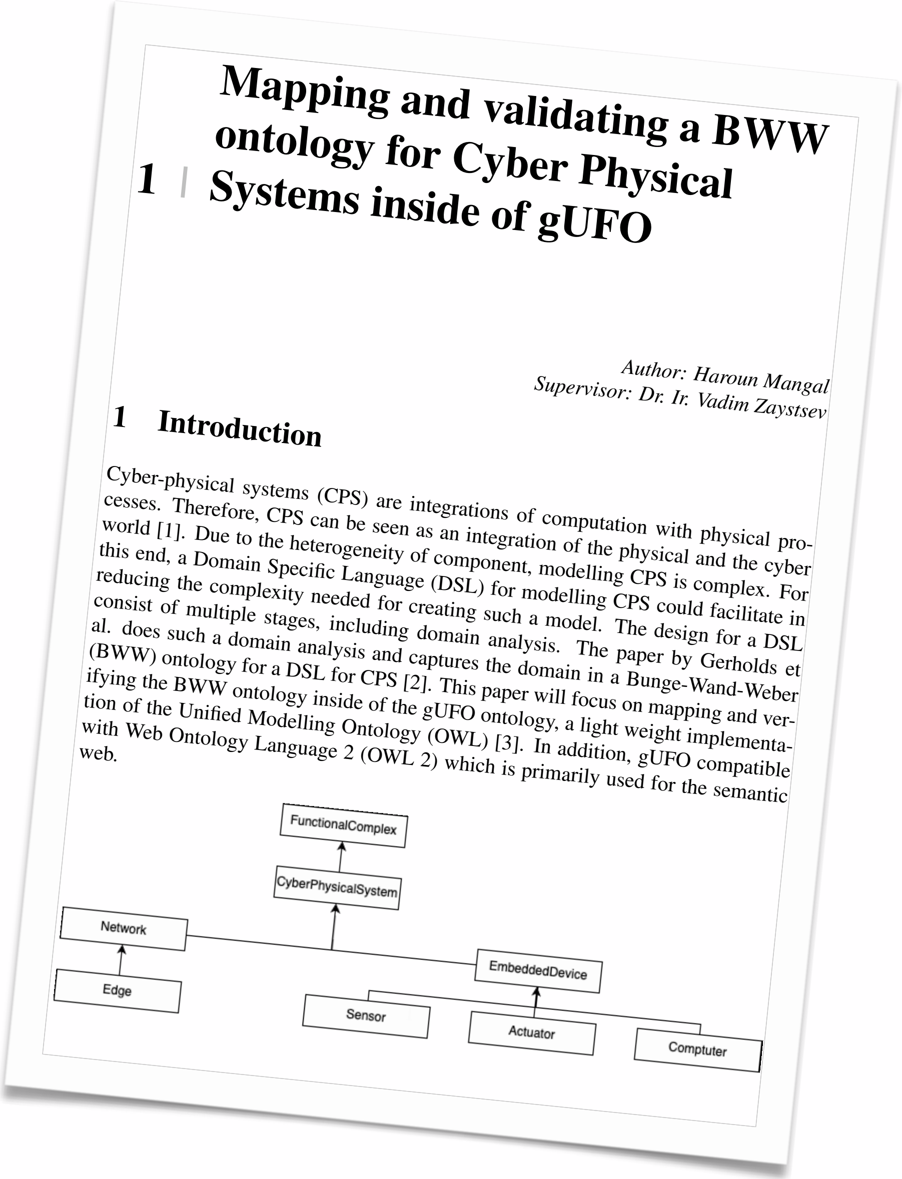
It is known from prior work that DSL development can be conceptually split into phases of **decision**, **analysis**, **design** and **implementation** [60]. By addressing these phases, we wish to provide a comprehensive framework for developing DSLs that can efficiently model CPS. Essentially we want the answers to the following research questions:

RQ1: What are critical aspects that DSLs must address to ensure that modelling and implementation of CPS to be most effective?
RQ2: How can we analyse and model the CPS domain to understand the foundational concepts and relationships that underpin these systems?

DOI: 10.1145/3652620.3688348

Analysis





1 | Mapping and validating a BWW ontology for Cyber Physical Systems inside of gUFO

Author: Haroun Mangal
Supervisor: Dr. Ir. Vadim Zaystsev

1 Introduction

Cyber-physical systems (CPS) are integrations of computation with physical processes. Therefore, CPS can be seen as an integration of the physical and the cyber world [1]. Due to the heterogeneity of component, modelling CPS is complex. For this end, a Domain Specific Language (DSL) for modelling CPS could facilitate in reducing the complexity needed for creating such a model. The design for a DSL consist of multiple stages, including domain analysis. The paper by Gerholds et al. does such a domain analysis and captures the domain in a Bunge-Wand-Weber (BWW) ontology for a DSL for CPS [2]. This paper will focus on mapping and verifying the BWW ontology inside of the gUFO ontology, a light weight implementation of the Unified Modelling Ontology (OWL) [3]. In addition, gUFO compatible with Web Ontology Language 2 (OWL 2) which is primarily used for the semantic web.

Design Phase

- **language exploitation**
  - piggyback, specialise, extend, invent
- **informal design**
  - text, natural models, examples of model instances
- **formal design**
  - grammars, abstract state machines
- **features**
  - what to include, how to support

Design

```
T>30 ; S=OK  ; E=0 ; D=high_temperature
T=_  ; S≠OK  ; E=0 ; D=sensor_failure
T=_  ; S=_   ; E=1 ; D=system_failure
```

```
normal → sensor_failure
    when E = 0 and S ≠ OK
normal → high_temperature
    when E = 0 and S = OK and T > 30
normal → system_error
    when E = 1
```

```
E = 1    ⇒ D = system_failure
S ≠ OK   ⇒ D = sensor_failure
T > 30   ⇒ D = high_temperature
```

DOI: [10.1145/3652620.3688348](https://doi.org/10.1145/3652620.3688348)

Conclusion

- making a **DSL** is a complex process
- **decision** phase: what's important?
- **analysis** phase: what's the domain?
- **design** phase: which features?
- we did the first steps for **CPS**

Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design

Marcus Gerhold
m.gerhold@utwente.nl
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Aliaksei Kouzel
a.kouzel@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Haroun Mangal
h.mangal@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Selin A. Mehmed
s.a.mehmed@student.utwente.nl
Technical Computer Science
University of Twente
Enschede, The Netherlands

Vadim Zaytsev
vadim@grammarware.net
Formal Methods and Tools
University of Twente
Enschede, The Netherlands

Abstract

Cyber-Physical Systems (CPS) integrate computational algorithms and physical components, requiring sophisticated modelling techniques to address complex interactions and dynamics. This paper explores the creation of Domain-Specific Languages (DSLs) tailored for CPS, focusing on the initial three critical phases: decision, analysis, design. We present four key aspects to address in the decision phase, design an ontology as a domain model for the analysis phase, and collect some advice for the design phase. By systematically addressing these phases, we provide a comprehensive framework for developing DSLs that can efficiently model CPS, facilitating improved design, verification, and deployment of these intricate systems.

CCS Concepts

• **Software and its engineering** → **Domain specific languages**; *Interoperability*; *Design languages*; • **Information systems** → **Ontologies**; • **Networks** → Network protocol design.

Keywords

Cyber-Physical Systems, Ontological Analysis, Domain-Specific Languages

ACM Reference Format:
Marcus Gerhold, Aliaksei Kouzel, Haroun Mangal, Selin A. Mehmed, and Vadim Zaytsev. 2024. Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688348

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3688348

1 Introduction

Cyber-Physical Systems (CPS) represent collaborations of computational algorithms and physical components [61], creating a network where digital systems monitor and control physical processes through sensors and actuators. These systems form a feedback loop where sensor data informs computational decisions, and actuators execute these decisions to affect the physical environment. Examples of CPS include smart grids, autonomous vehicles, and advanced medical monitoring systems [66, 83]. CPS hold substantial potential across diverse domains such as smart manufacturing, robotics, healthcare, intelligent transportation, and smart cities, offering benefits like enhanced automation, improved safety, and optimised resource usage [35, 42, 61, 72].

Realising the full potential of CPS poses significant challenges due to their inherent complexity and heterogeneity [83]. Integrating the continuous, concurrent physical world with the discrete, sequential cyber world often leads to non-deterministic behaviours, complicating the development of reliable and dependable models. Model-Driven Engineering (MDE) addresses these complexities by breaking down CPS into manageable components, yet the intricacies of such systems demand specialised modelling approaches [22].

Domain-Specific Languages (DSLs) offer a promising solution for CPS modelling by providing tailored notations and constructs specific to the domain [23, 87]. Unlike general-purpose languages (GPLs), DSLs can encapsulate domain-specific knowledge, simplifying the modelling process and enhancing communication between domain experts and developers [59]. For instance, UML and VHDL are well-known DSLs in their respective domains.

It is known from prior work that DSL development can be conceptually split into phases of **decision**, **analysis**, **design** and **implementation** [60]. By addressing these phases, we wish to provide a comprehensive framework for developing DSLs that can efficiently model CPS. Essentially we want the answers to the following research questions:

RQ1: What are critical aspects that DSLs must address to ensure that modelling and implementation of CPS to be most effective?

RQ2: How can we analyse and model the CPS domain to understand the foundational concepts and relationships that underpin these systems?

Zorro

Zero downtime in cyber physical systems

https://zorro-project.nl/

UNIVERSITY OF TWENTE. | FMT Formal Methods & Tools