EUROPEAN SMALLTALK USER GROUP

esug

www.esug.org

SoSAF: A Pharo-Based Framework for Enhancing System-Of-Systems Dependencies Analysis

Mouhamadou F. BALL - Patrick AUGER

ENSTA BRETAGNE

UNIVERSITÉ LUMIÈRE LYON 2

Towards System of Systems Architecture (SoS)

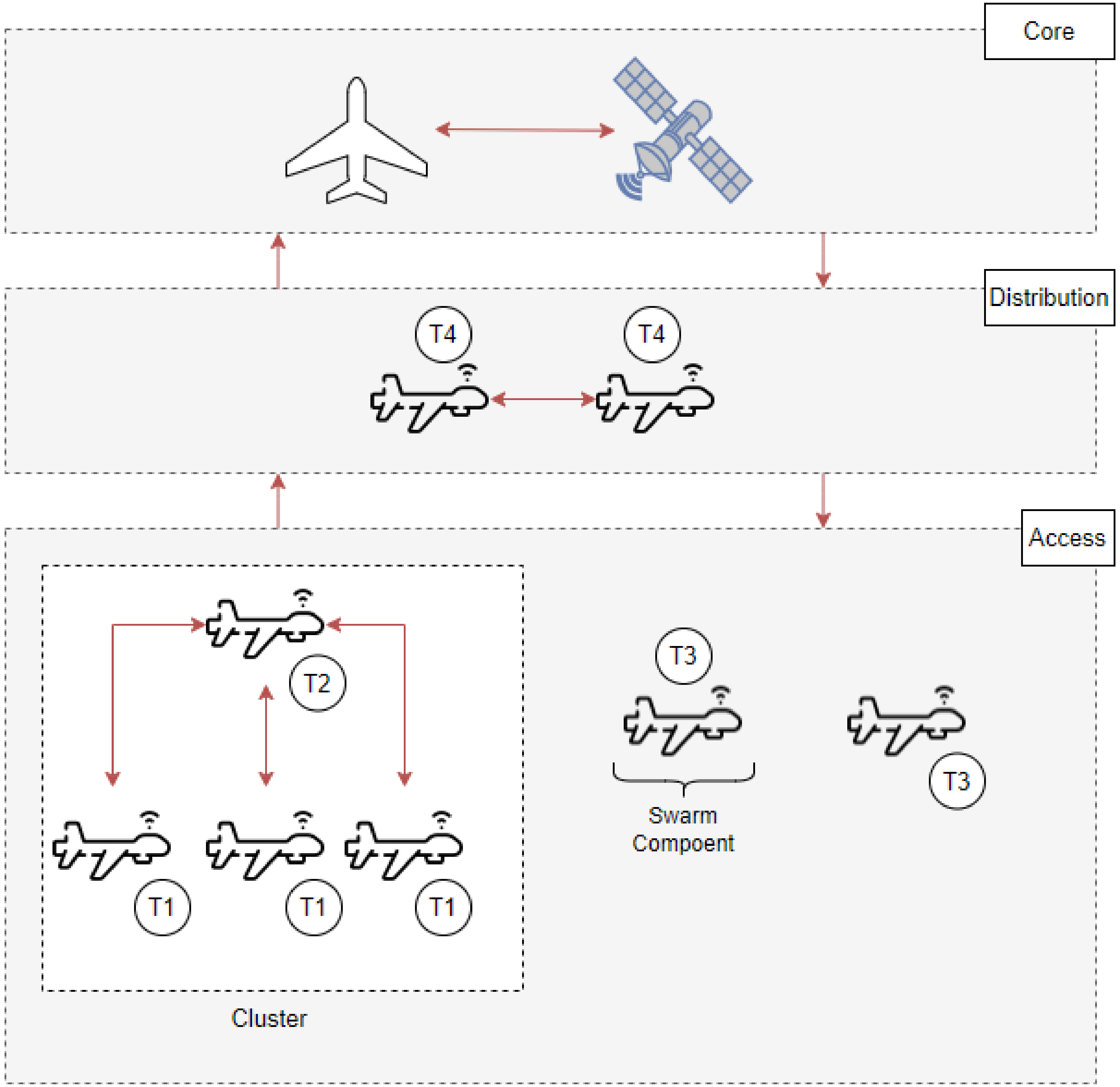
System of Systems:
finite number of constituent systems which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal.

Maier (1998) [1]

Tasks - Missions:

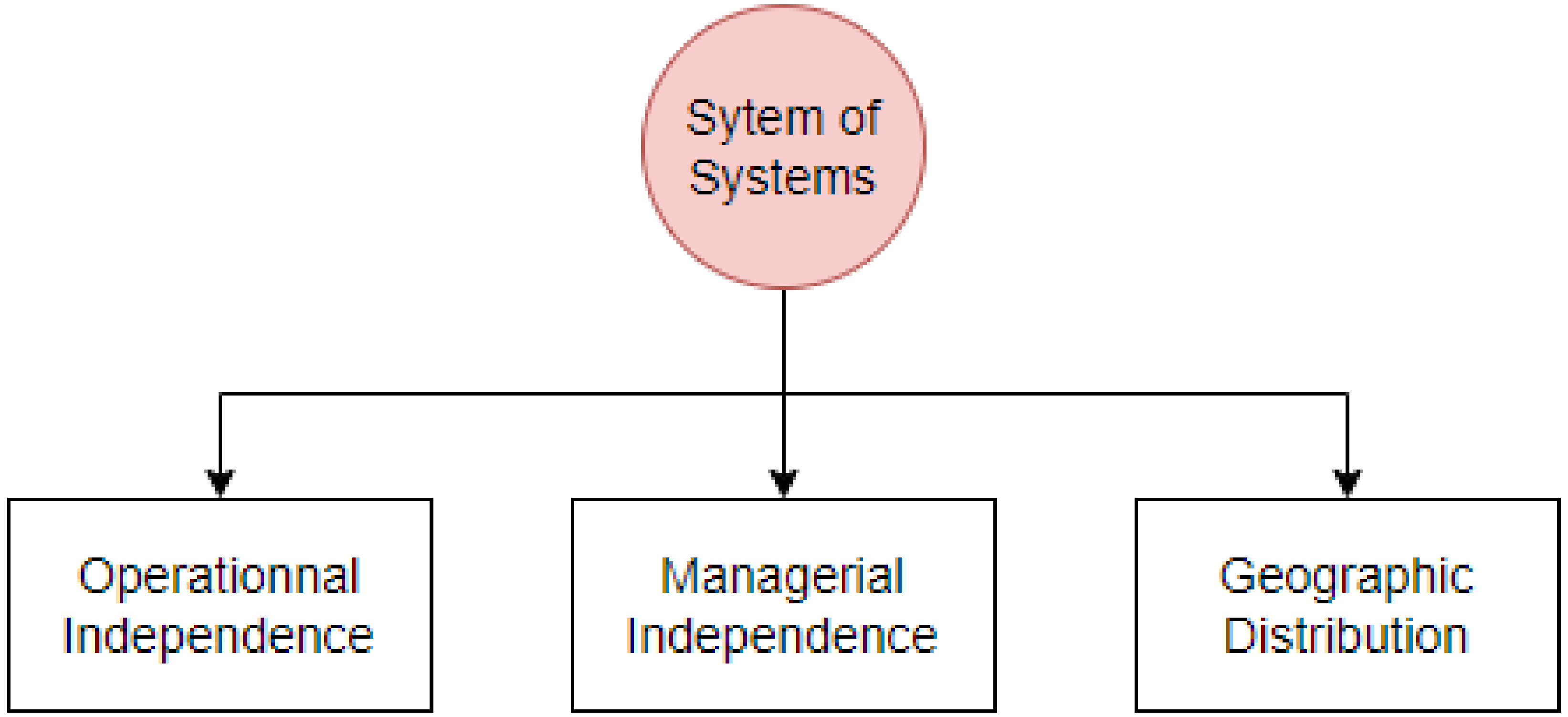
$$\text{let } T = \{T_1, T_2, \dots, T_n\} \quad \text{and} \quad M = \{M_1, M_2, \dots, M_m\}$$

$$\forall M_i \in M, \quad M_i \subseteq T$$



System of Systems example : Unmanned Vehicules (UVs)

System of Systems Specificities

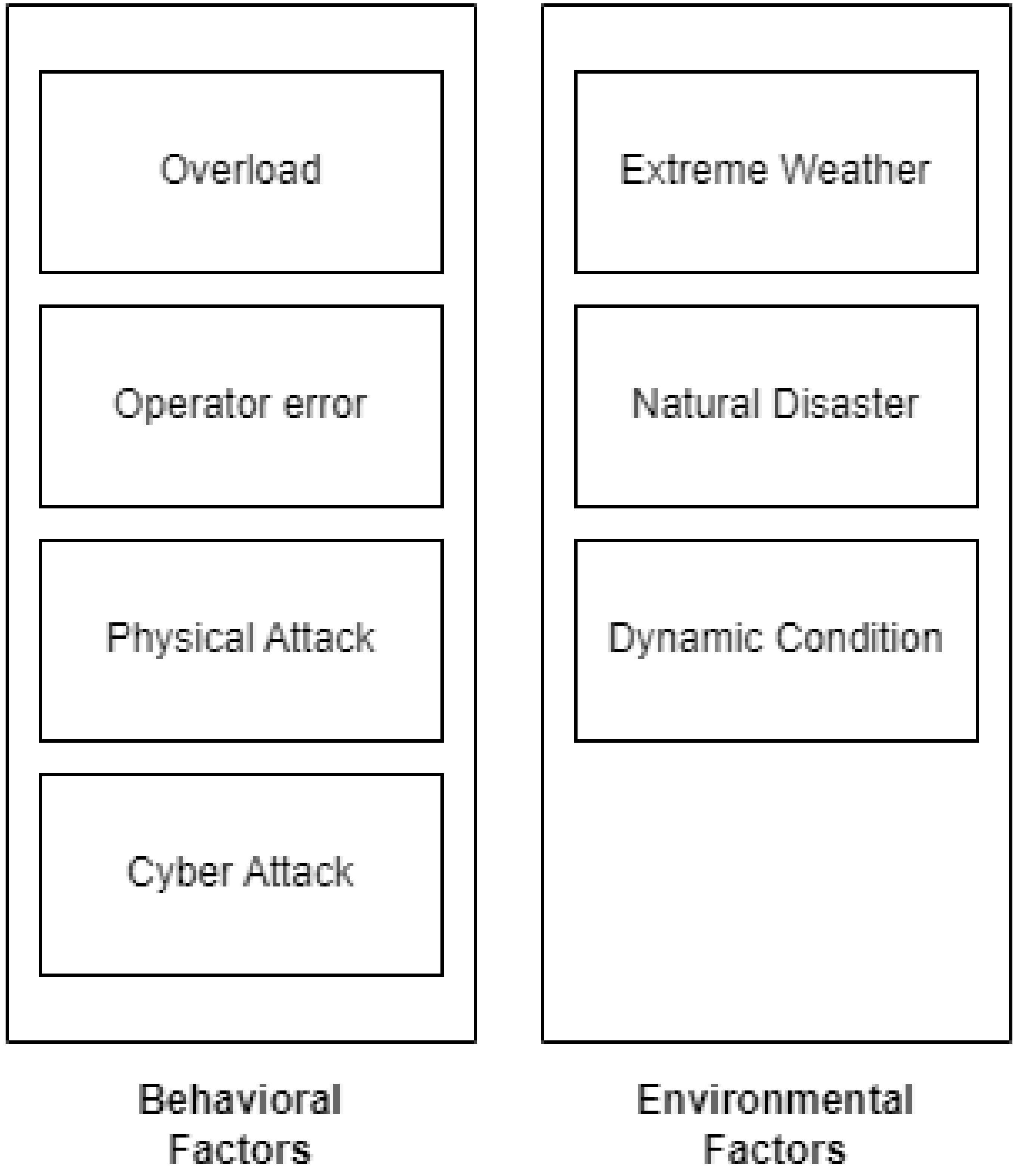


System of Systems properties

Maier (1998) [1]

- Scalability
  - Horizontal
- Flexibility and Adaptability
  - Modular Approach
- Robustess and Reliability
  - Fault Tolerance
  - Redundancy
- Enhanced Performance
  - Load Distribution
  - Specialized System

System of Systems facing hazard



Overload

Operator error

Physical Attack

Cyber Attack

Behavioral Factors

Extreme Weather

Natural Disaster

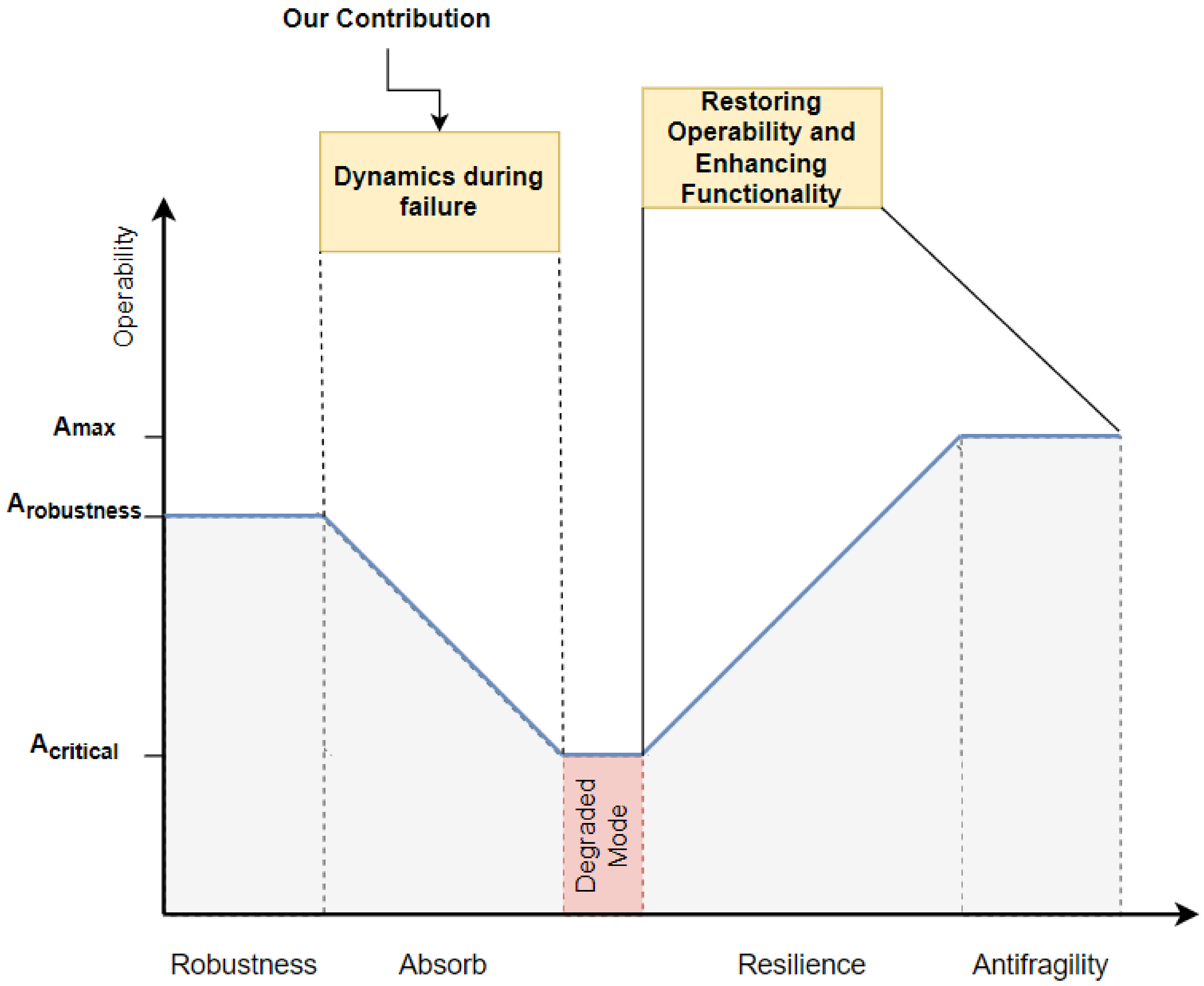
Dynamic Condition

Environmental Factors

Cascading failures main causes

Xing (2020) [2]

System of Systems Dynamics During Failure



System operability during failure

Main issues :

- **Dynamics during failure:** Evaluate system operability during failures through SoS configuration
- **Restoring Operability and Enhancing Features**

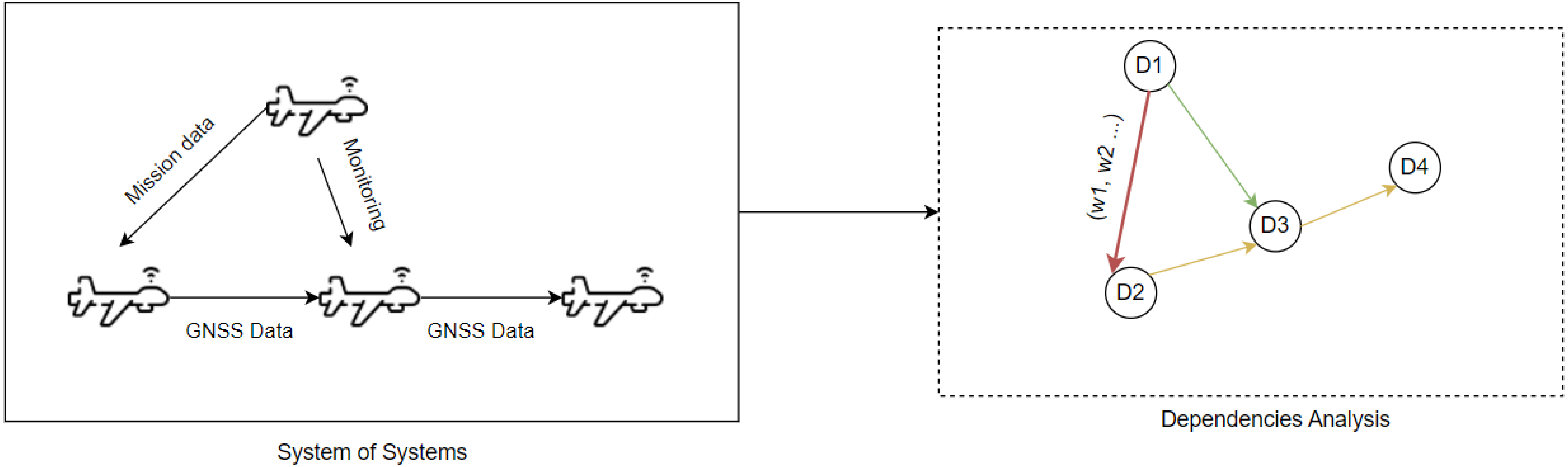
Why Dependency Analysis ?

Dynamics during failure

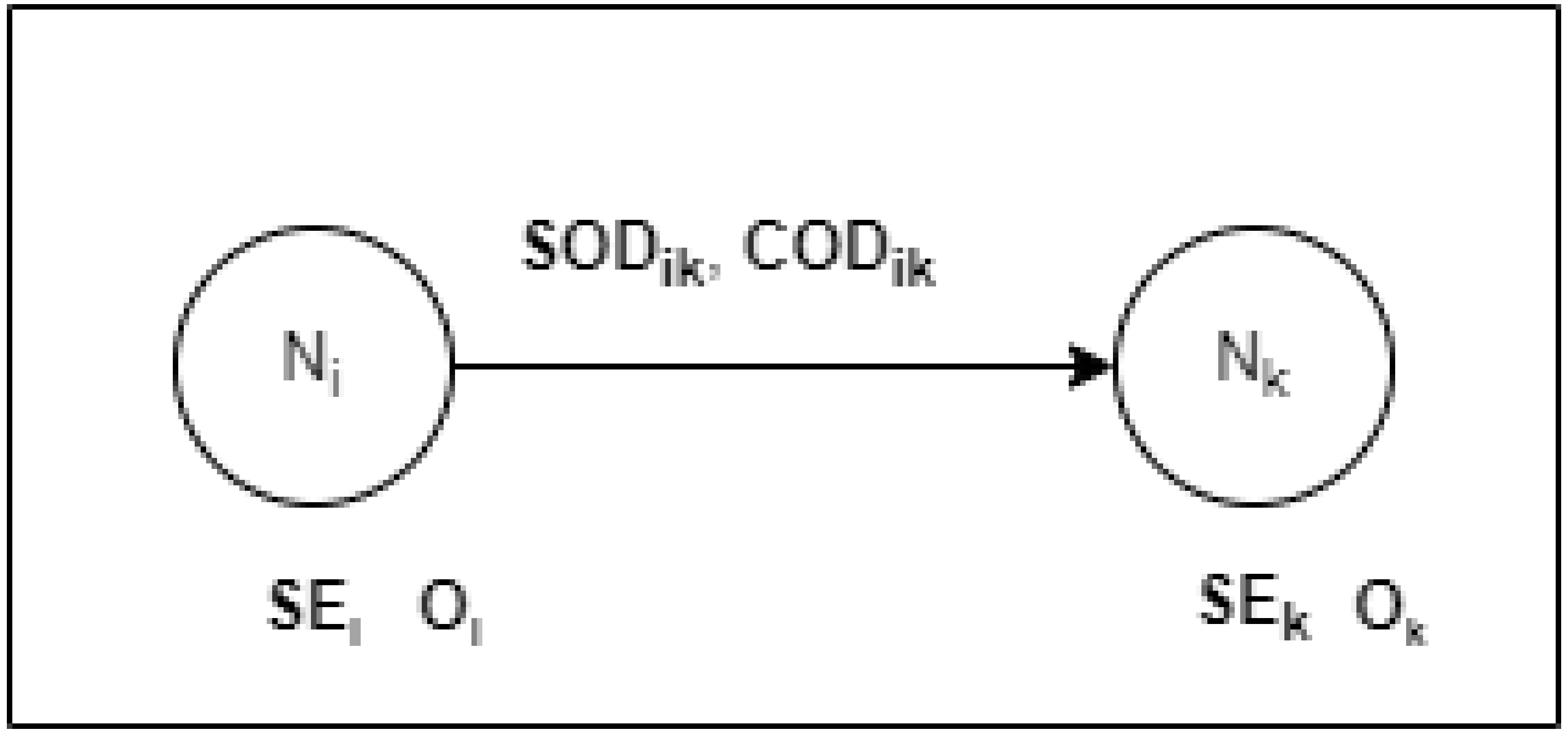
Dependency Analysis

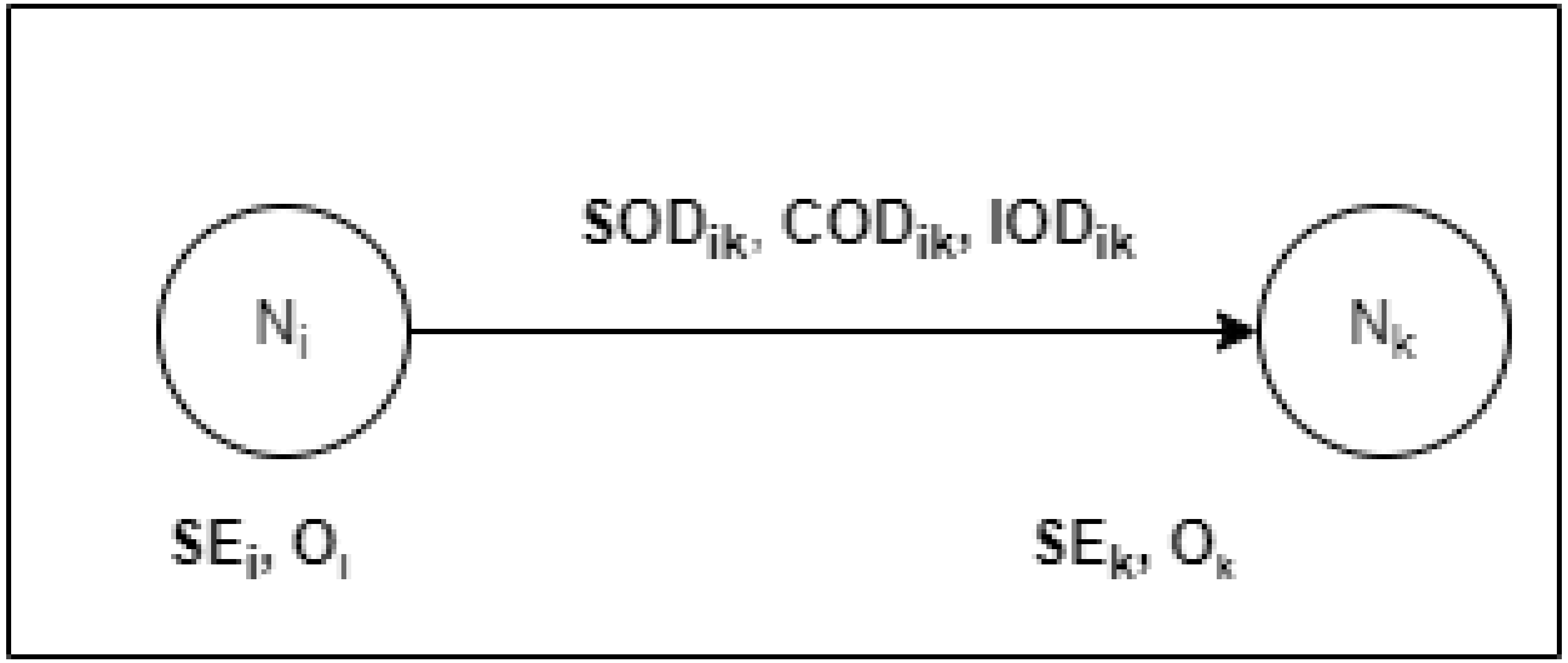
- Model SoS
    - Dependency graph
    - Mathematical Model
- Analyze the Dependencies Network
- Identify Critical Dependencies and Bottlenecks

Dependency Analysis



Dynamics During Failure: Existing Works on Dependencies Graph





Functional Dependencies Network Analysis (FDNA)

- SOD : Strength of Dependency
- COD : Critical of Dependency

Garvey et al. (2009) [3]

System Operational Dependencies Analysis (SODA)

- IOD : Impact of Dependency

Guariniello et al. (2017) [4]

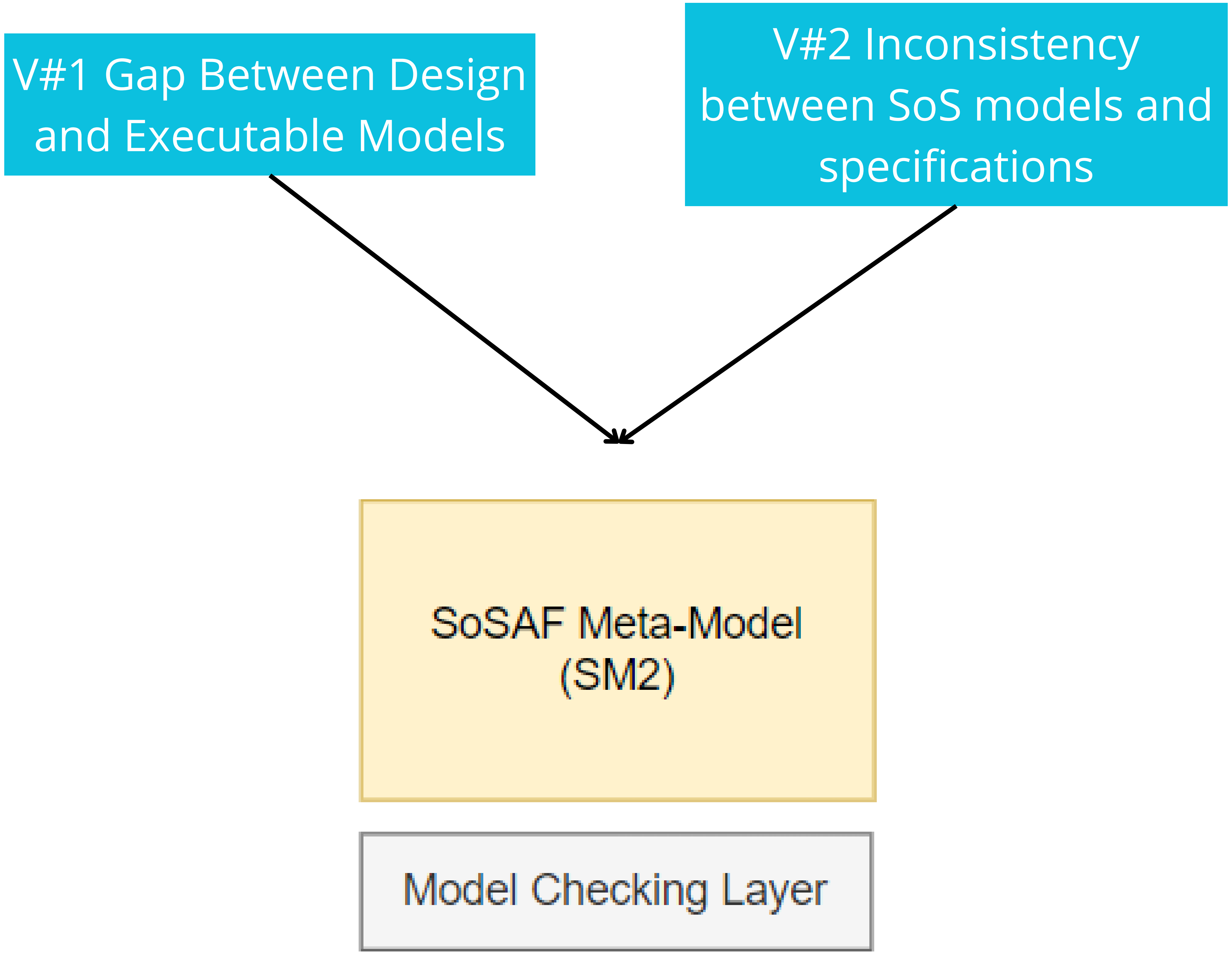
Scientific identified locks

V#1 - Gap Between Design and Executable Models
There is a disconnect between the high-level design models and the low-level executable models. This gap can lead to inconsistencies and errors when transitioning from design to implementation.

V#2 - Inconsistency between SoS models and specifications
The absence of a formal model checking layer means that the proposed models are not rigorously verified against desired properties and specifications.

V#3 - Lack of Interactive SoS Simulation:
Limits the ability to test and validate system behavior in a dynamic and iterative manner. Interactive simulation allows for real-time exploration and debugging of the model.

Contributions to Solving Current Problems



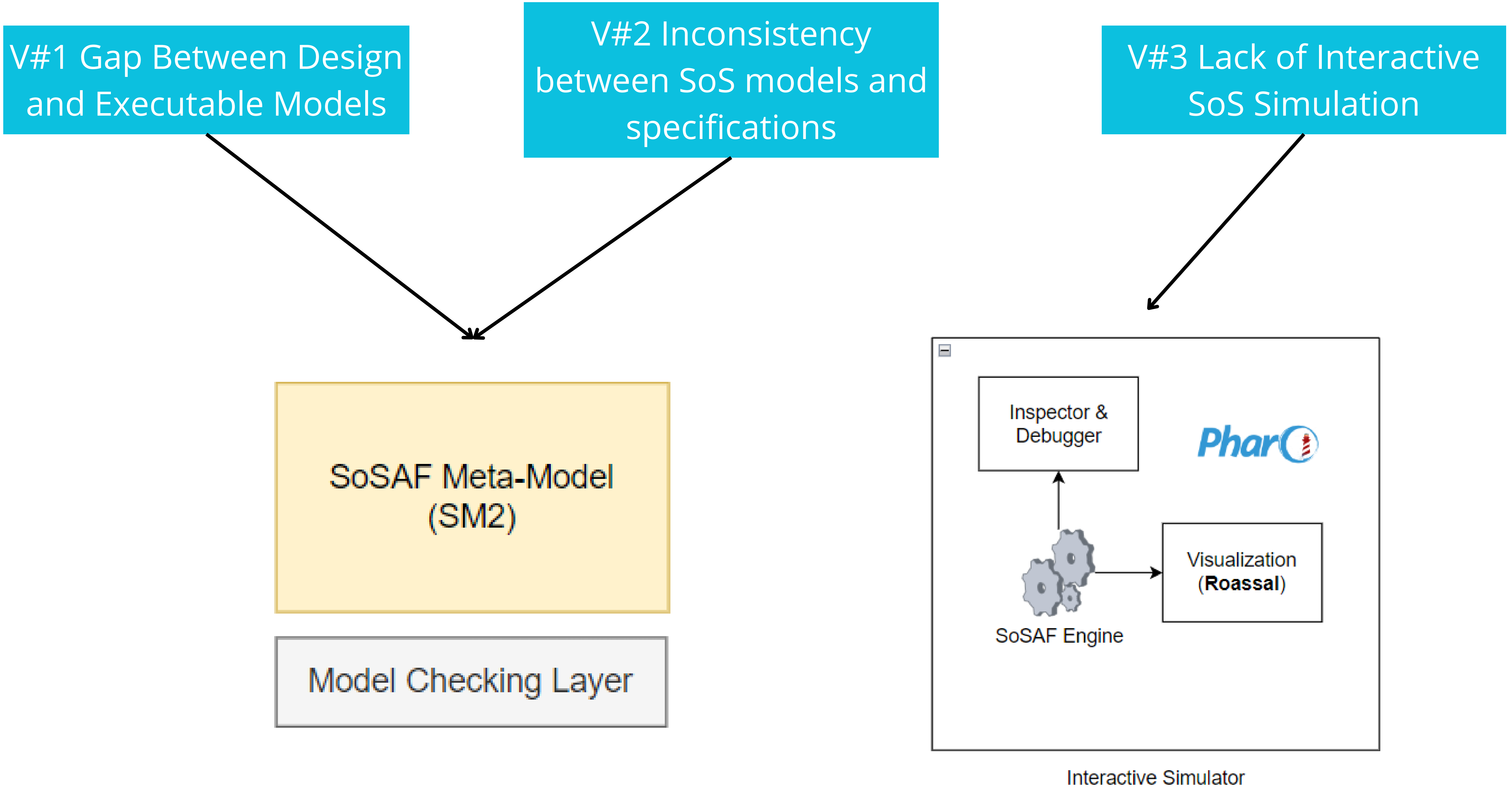
V#1 Gap Between Design and Executable Models

V#2 Inconsistency between SoS models and specifications

SoSAF Meta-Model (SM2)

Model Checking Layer

Contributions to Solving Current Problems



V#1 Gap Between Design and Executable Models

V#2 Inconsistency between SoS models and specifications

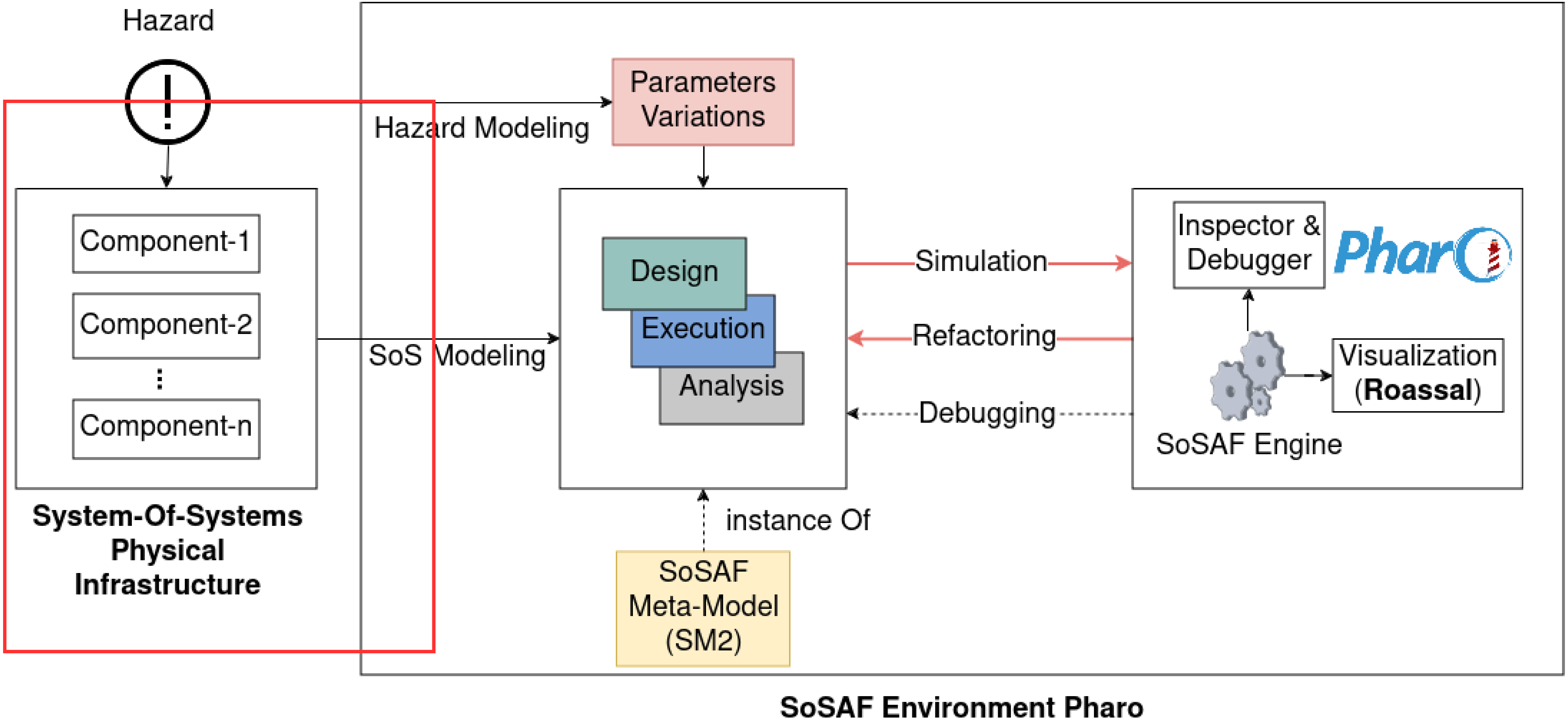
V#3 Lack of Interactive SoS Simulation

SoSAF Meta-Model (SM2)

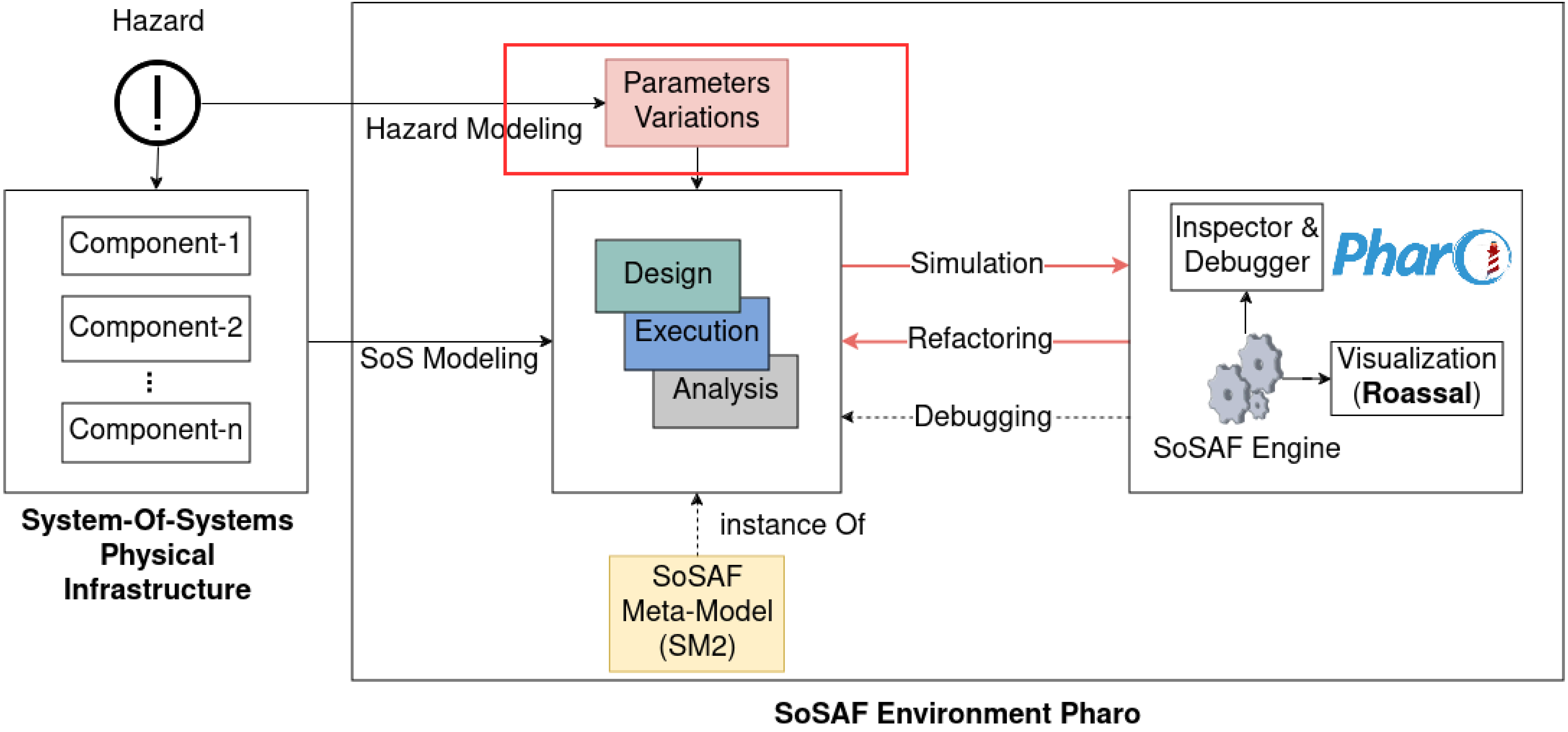
Model Checking Layer

Interactive Simulator

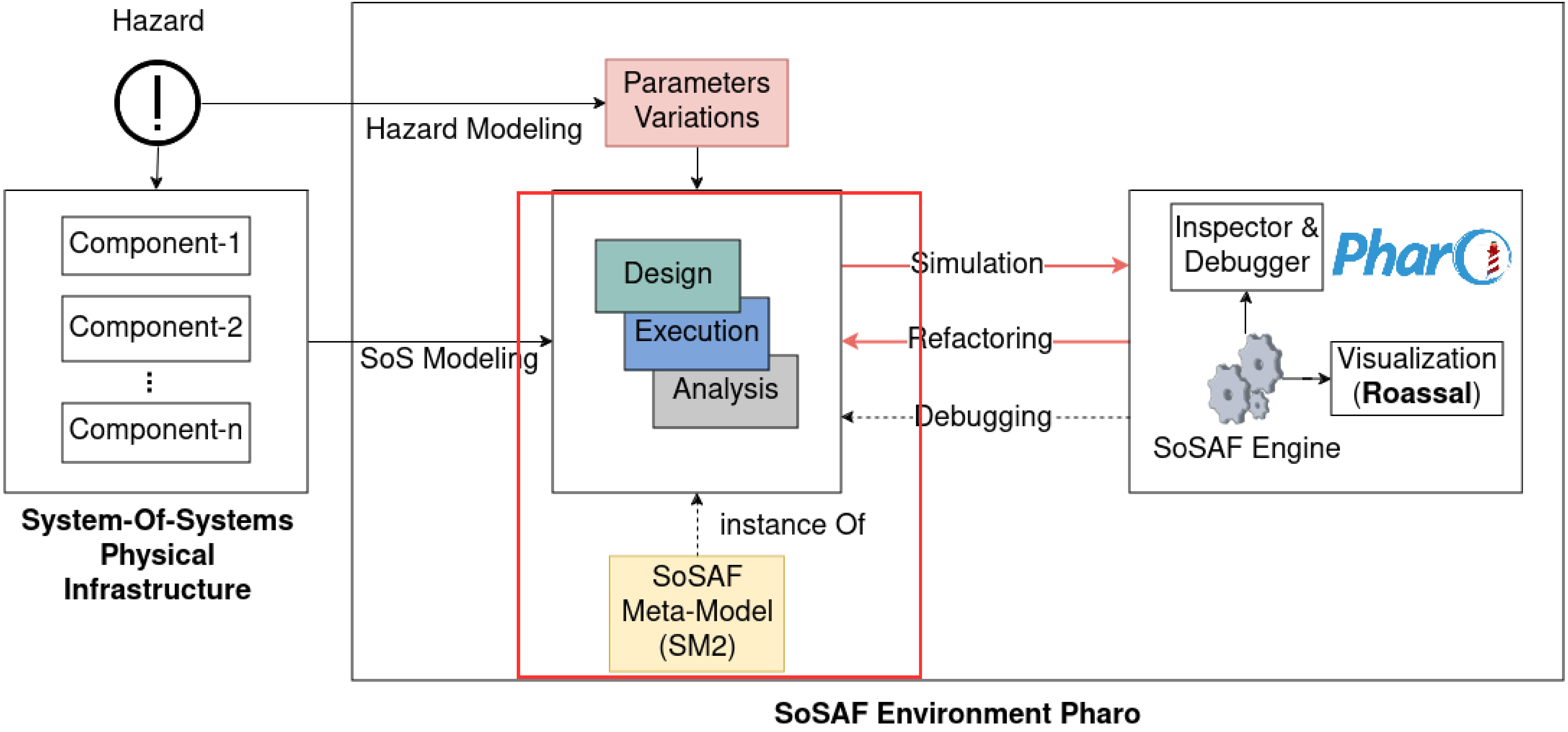
SoSAF: System of Systems Architecture Framework (1/4)



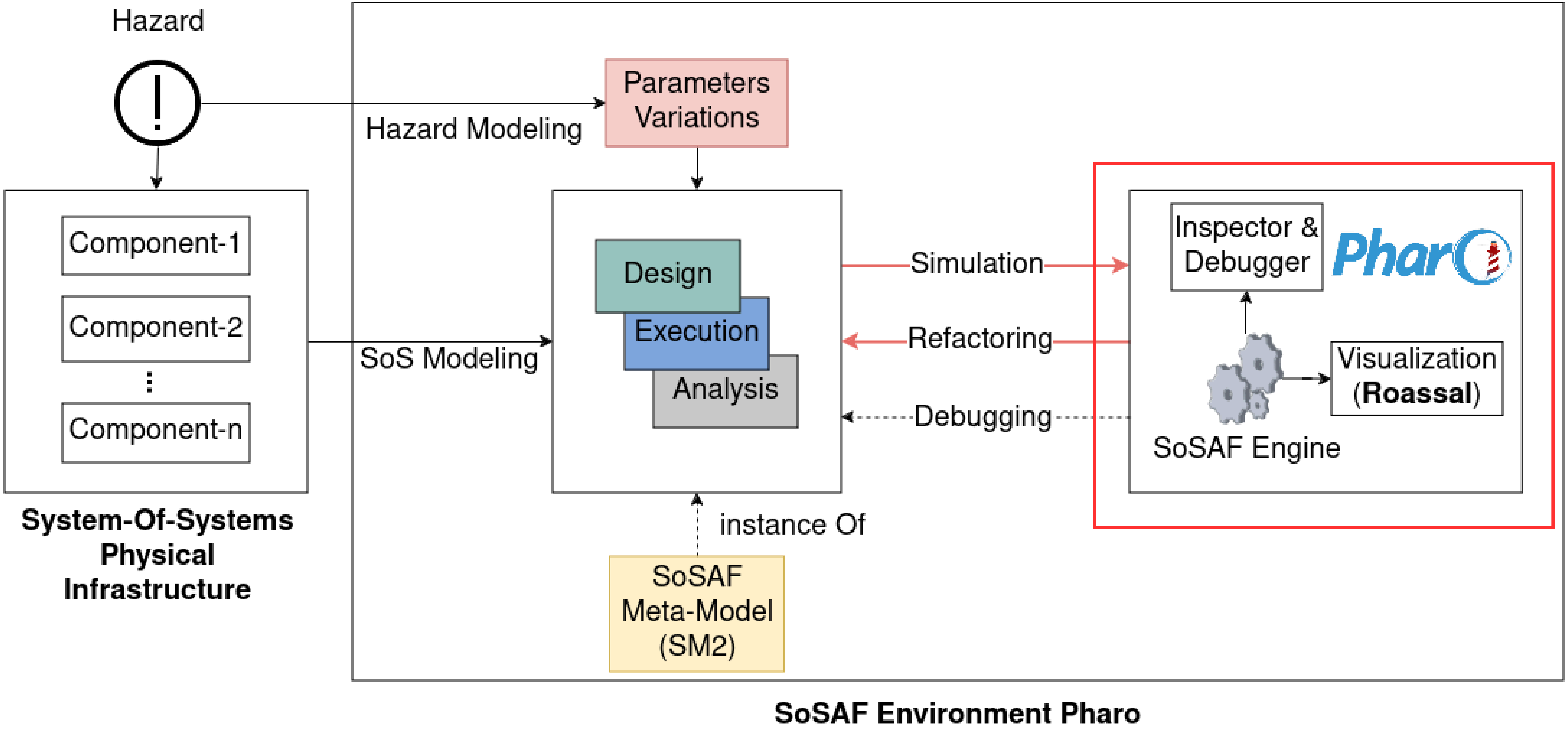
SoSAF: System of Systems Architecture Framework (1/4)



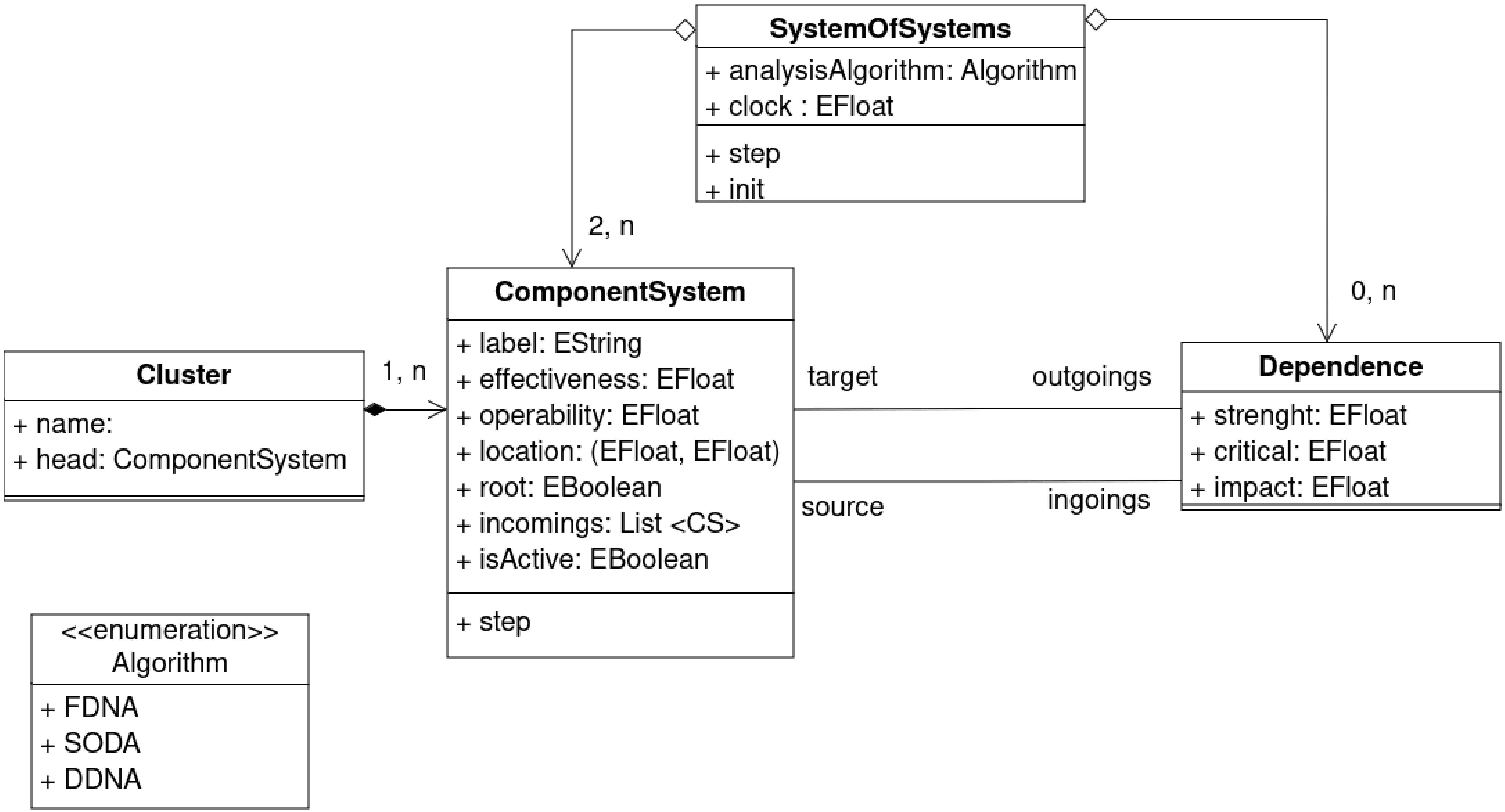
SoSAF: System of Systems Architecture Framework (1/4)



SoSAF: System of Systems Architecture Framework (1/4)

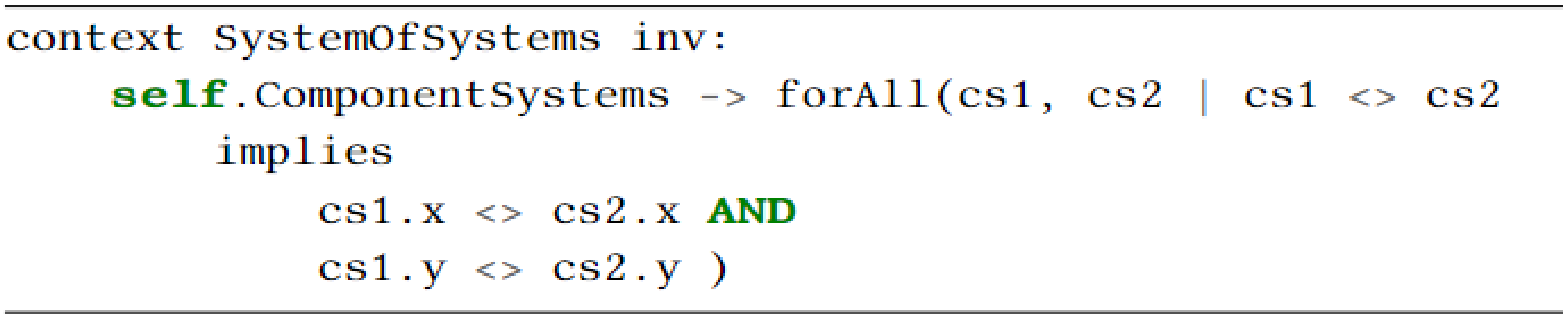


SoSAF: System of Systems Architecture Framework (2/4) - MOF



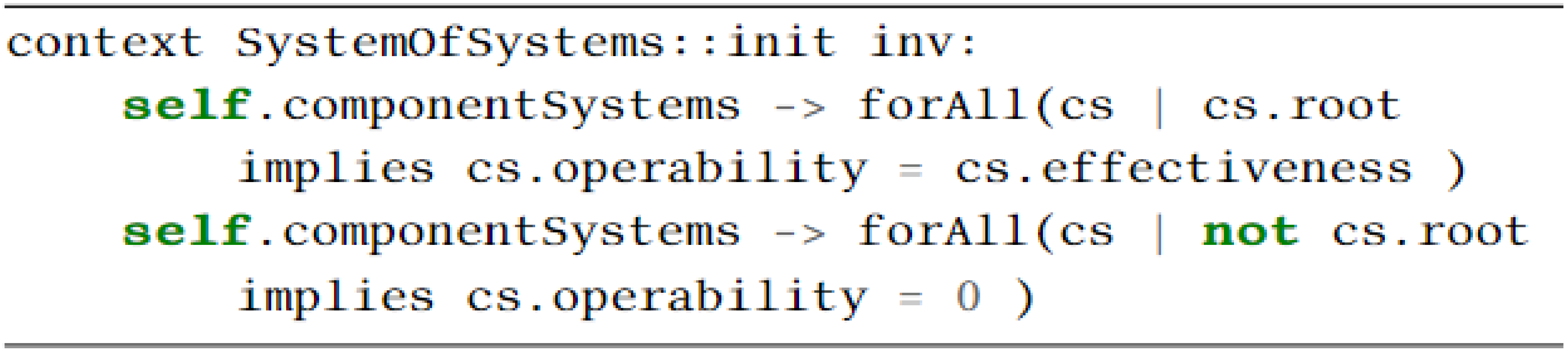
SoSAF: System of Systems Architecture Framework (3/4) - OCL

1- Location Constraint



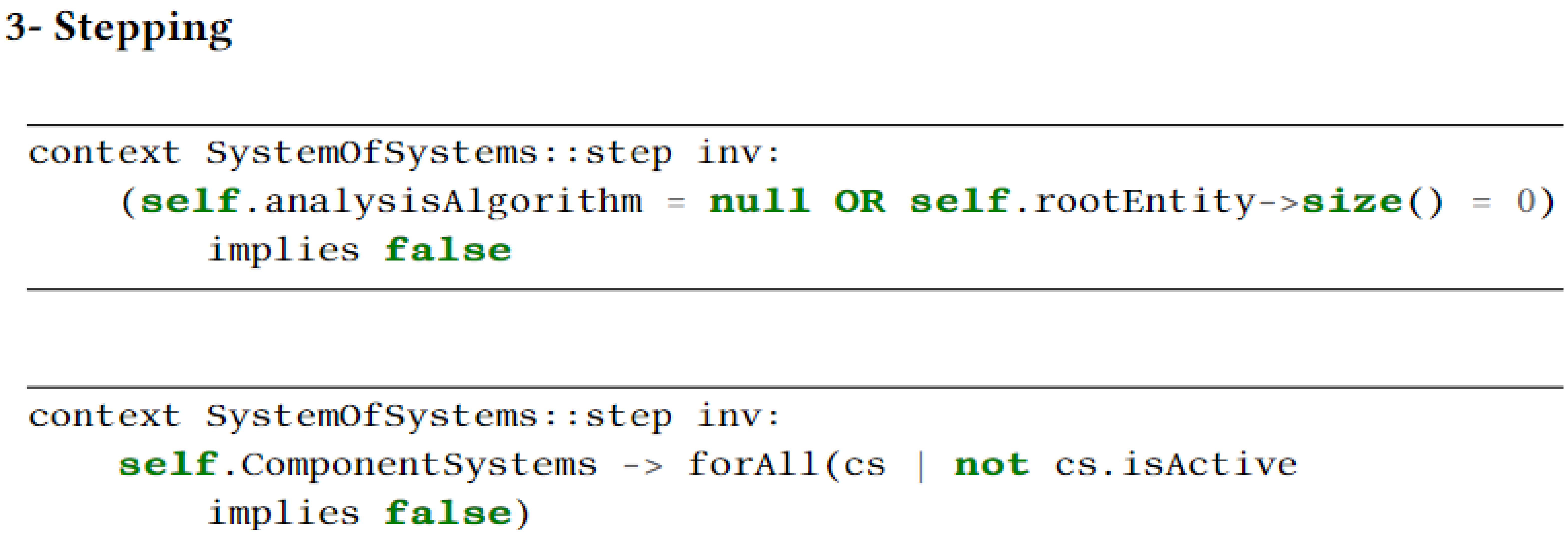
```
context SystemOfSystems inv:
    self.ComponentSystems -> forAll(cs1, cs2 | cs1 <> cs2
        implies
            cs1.x <> cs2.x AND
            cs1.y <> cs2.y )
```

2- Initialization



```
context SystemOfSystems::init inv:
    self.componentSystems -> forAll(cs | cs.root
        implies cs.operability = cs.effectiveness )
    self.componentSystems -> forAll(cs | not cs.root
        implies cs.operability = 0 )
```

SoSAF: System of Systems Architecture Framework (4/4) - OCL

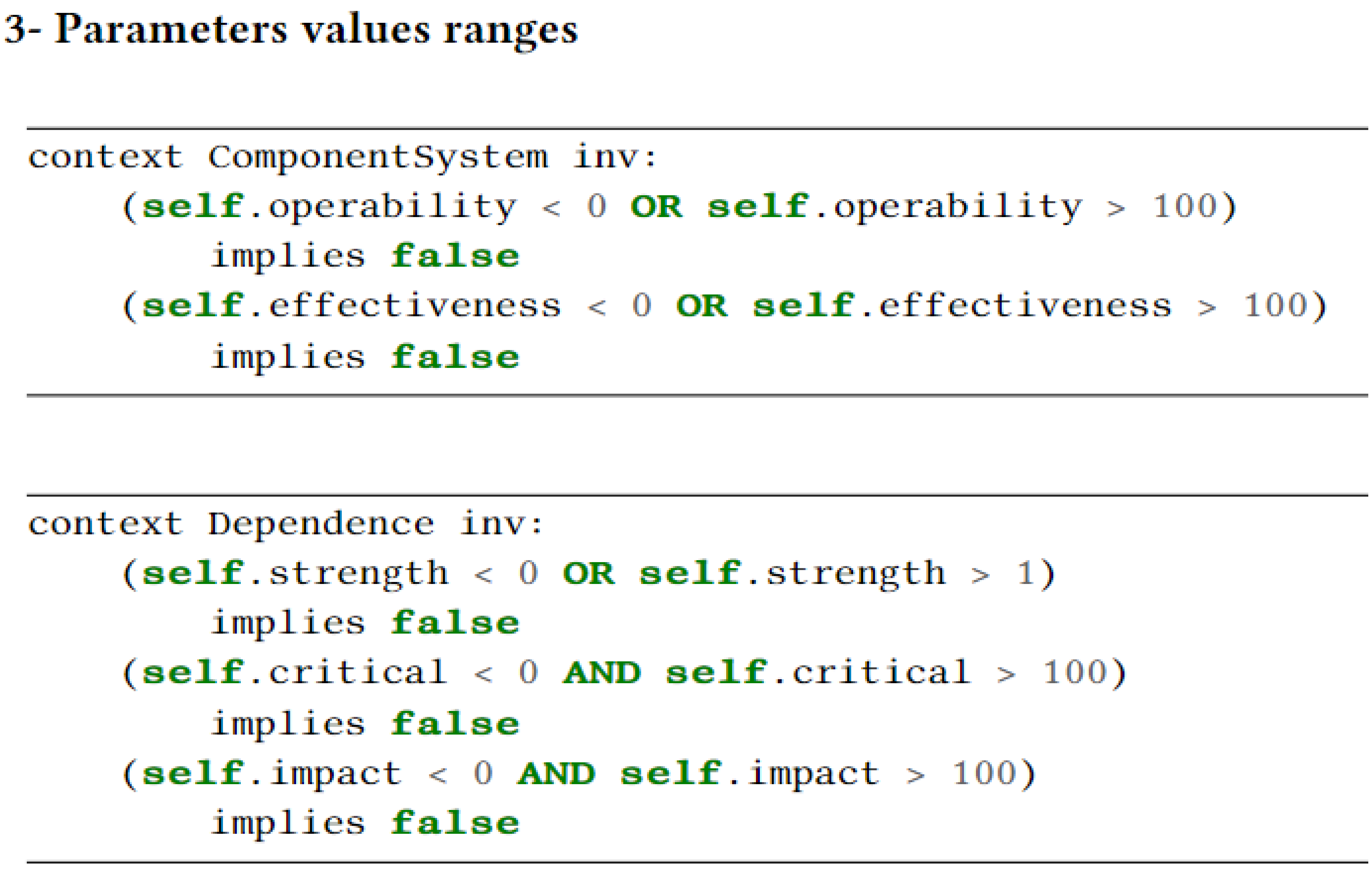


3- Stepping

```
context SystemOfSystems::step inv:
    (self.analysisAlgorithm = null OR self.rootEntity->size() = 0)
        implies false
```

```
context SystemOfSystems::step inv:
    self.ComponentSystems -> forAll(cs | not cs.isActive
        implies false)
```

SoSAF: System of Systems Architecture Framework (4/4) - OCL

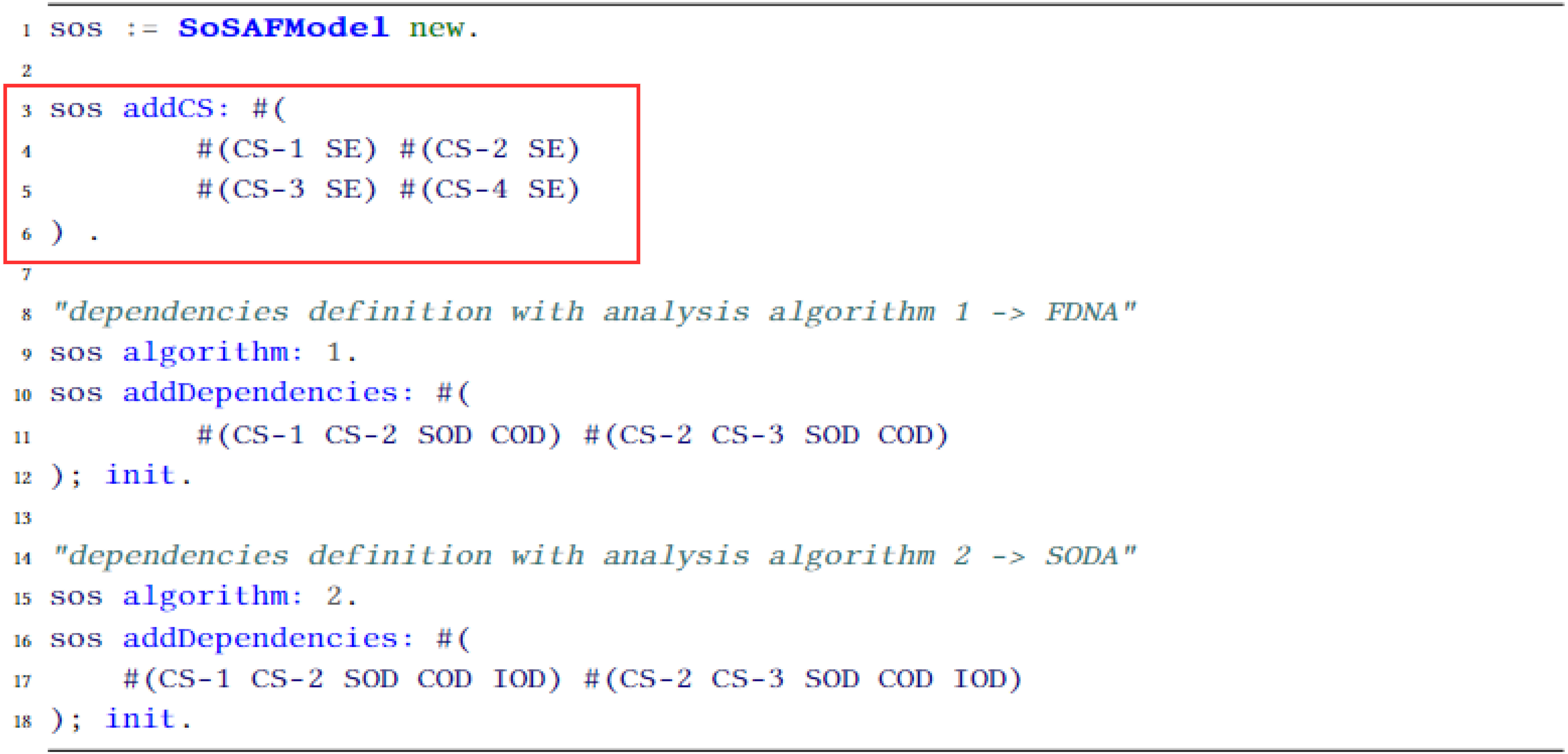


3- Parameters values ranges

```
context ComponentSystem inv:
    (self.operability < 0 OR self.operability > 100)
        implies false
    (self.effectiveness < 0 OR self.effectiveness > 100)
        implies false
```

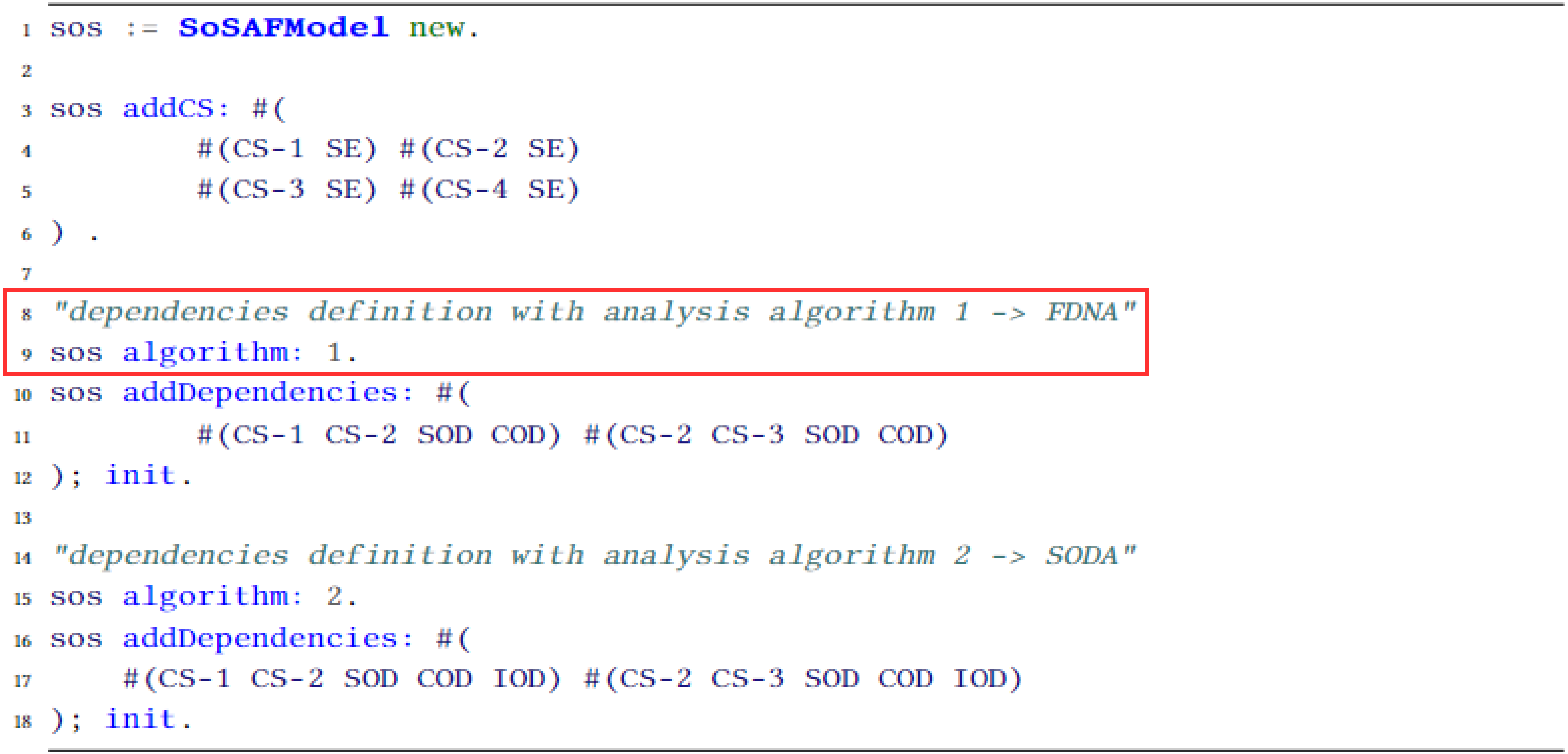
```
context Dependence inv:
    (self.strength < 0 OR self.strength > 1)
        implies false
    (self.critical < 0 AND self.critical > 100)
        implies false
    (self.impact < 0 AND self.impact > 100)
        implies false
```

SoSAF: SM2 instance with Pharo language



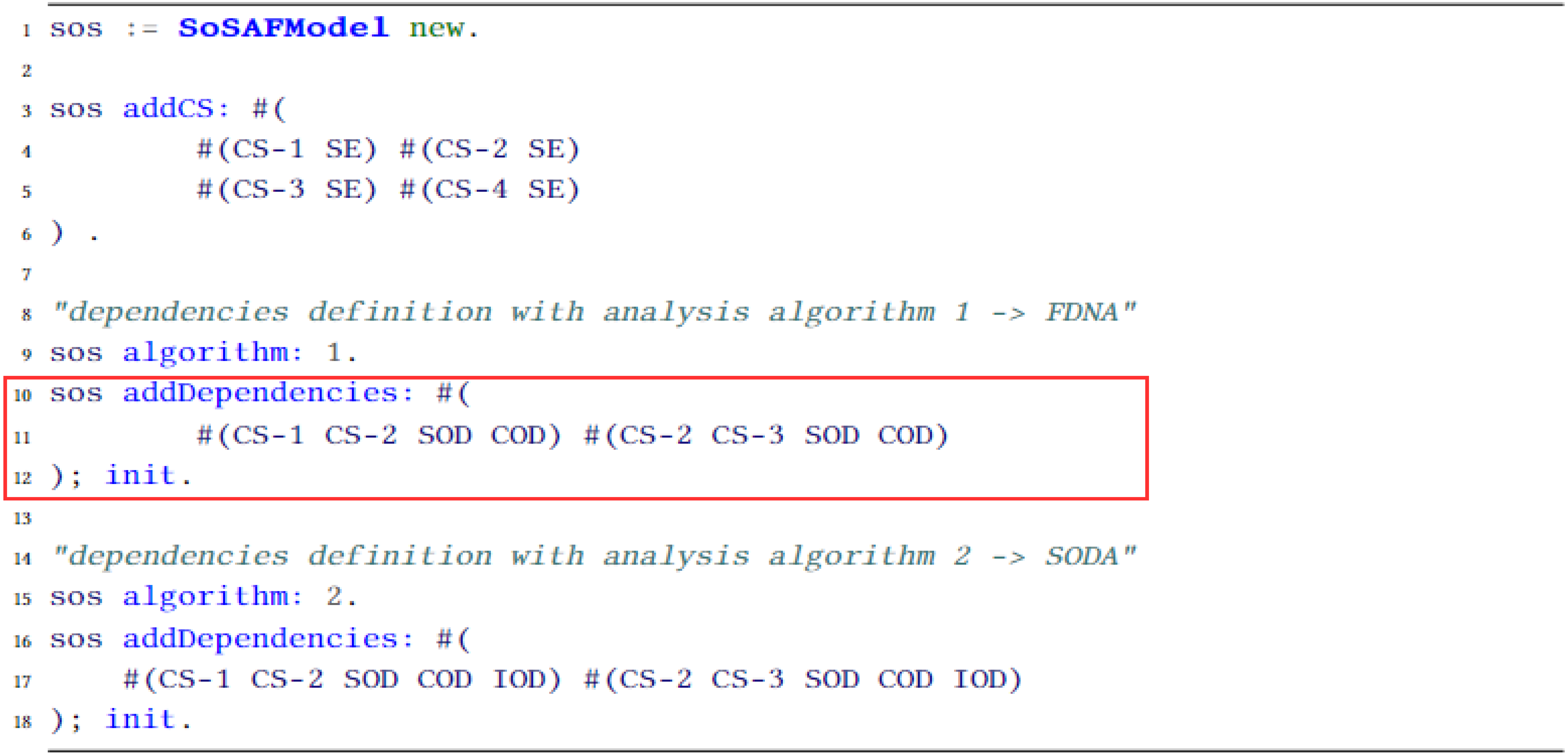
```
sos := SoSAFModel new.

sos addCS: #(
        #(CS-1 SE) #(CS-2 SE)
        #(CS-3 SE) #(CS-4 SE)
) .

"dependencies definition with analysis algorithm 1 -> FDNA"
sos algorithm: 1.
sos addDependencies: #(
        #(CS-1 CS-2 SOD COD) #(CS-2 CS-3 SOD COD)
); init.

"dependencies definition with analysis algorithm 2 -> SODA"
sos algorithm: 2.
sos addDependencies: #(
    #(CS-1 CS-2 SOD COD IOD) #(CS-2 CS-3 SOD COD IOD)
); init.
```

SoSAF: SM2 instance with Pharo language



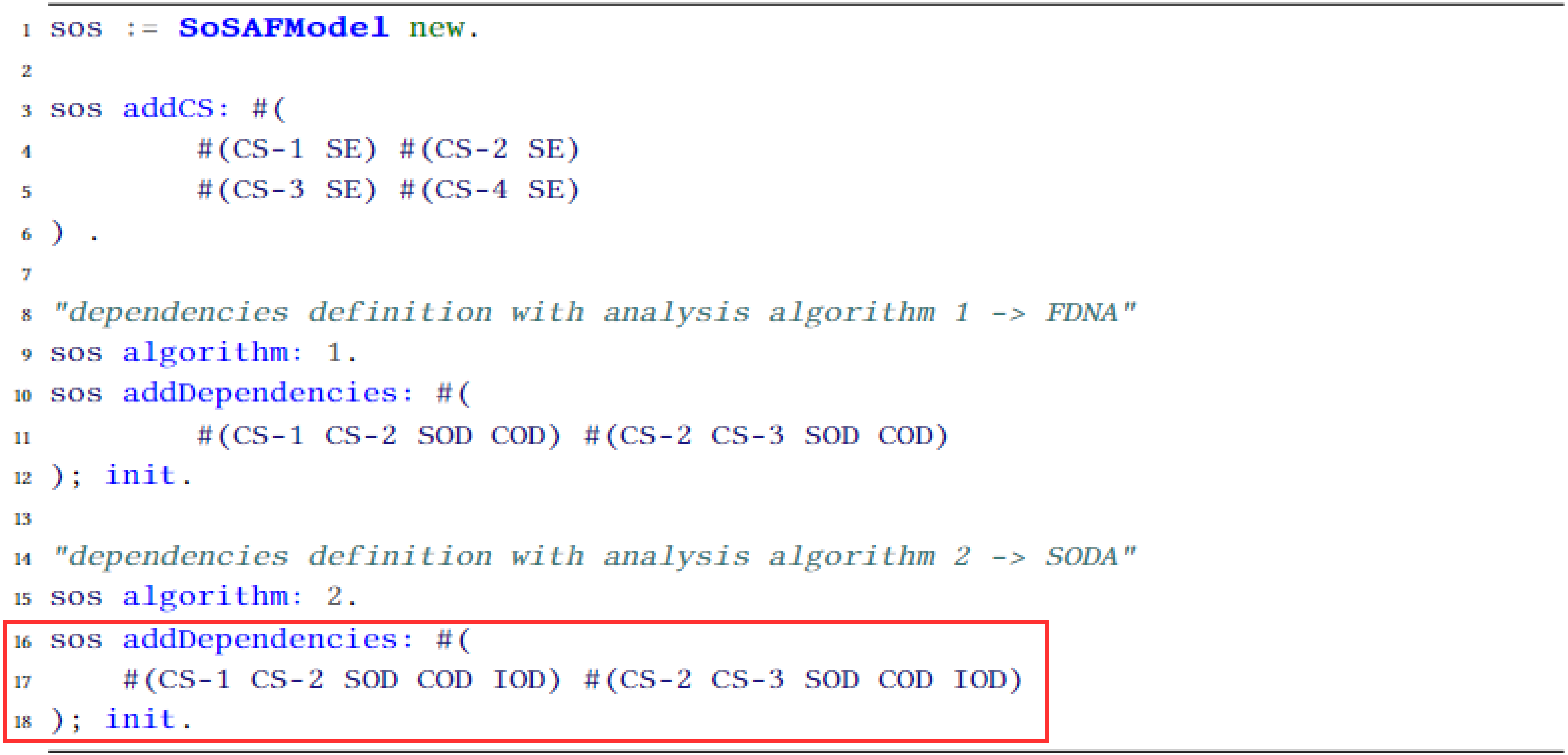
```
sos := SoSAFModel new.

sos addCS: #(
        #(CS-1 SE) #(CS-2 SE)
        #(CS-3 SE) #(CS-4 SE)
) .

"dependencies definition with analysis algorithm 1 -> FDNA"
sos algorithm: 1.
sos addDependencies: #(
        #(CS-1 CS-2 SOD COD) #(CS-2 CS-3 SOD COD)
); init.

"dependencies definition with analysis algorithm 2 -> SODA"
sos algorithm: 2.
sos addDependencies: #(
    #(CS-1 CS-2 SOD COD IOD) #(CS-2 CS-3 SOD COD IOD)
); init.
```

SoSAF: SM2 instance with Pharo language



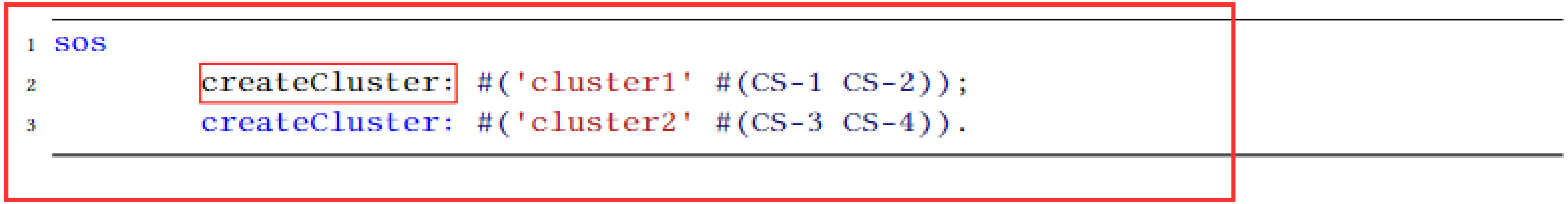
```
sos := SoSAFModel new.

sos addCS: #(
        #(CS-1 SE) #(CS-2 SE)
        #(CS-3 SE) #(CS-4 SE)
) .

"dependencies definition with analysis algorithm 1 -> FDNA"
sos algorithm: 1.
sos addDependencies: #(
        #(CS-1 CS-2 SOD COD) #(CS-2 CS-3 SOD COD)
); init.

"dependencies definition with analysis algorithm 2 -> SODA"
sos algorithm: 2.
sos addDependencies: #(
    #(CS-1 CS-2 SOD COD IOD) #(CS-2 CS-3 SOD COD IOD)
); init.
```

SoSAF: SM2 instance with Pharo language

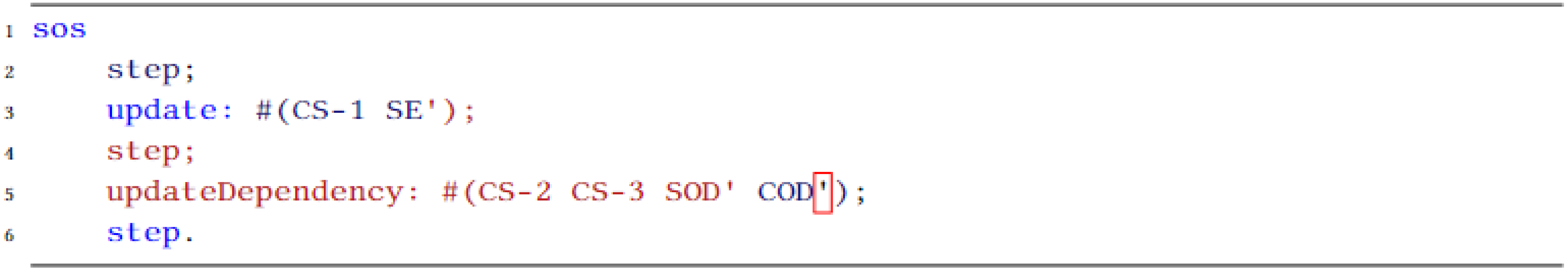


```
sos := SoSAFModel new.

sos addCS: #(
        #(CS-1 SE) #(CS-2 SE)
        #(CS-3 SE) #(CS-4 SE)
) .

"dependencies definition with analysis algorithm 1 -> FDNA"
sos algorithm: 1.
sos addDependencies: #(
        #(CS-1 CS-2 SOD COD) #(CS-2 CS-3 SOD COD)
); init.

"dependencies definition with analysis algorithm 2 -> SODA"
sos algorithm: 2.
sos addDependencies: #(
    #(CS-1 CS-2 SOD COD IOD) #(CS-2 CS-3 SOD COD IOD)
); init.
```

SoSAF: SM2 instance with Pharo language

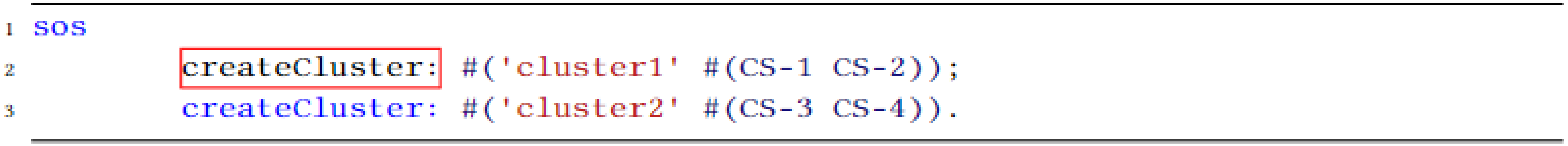


```
sos
        createCluster: #('cluster1' #(CS-1 CS-2));
        createCluster: #('cluster2' #(CS-3 CS-4)).
```

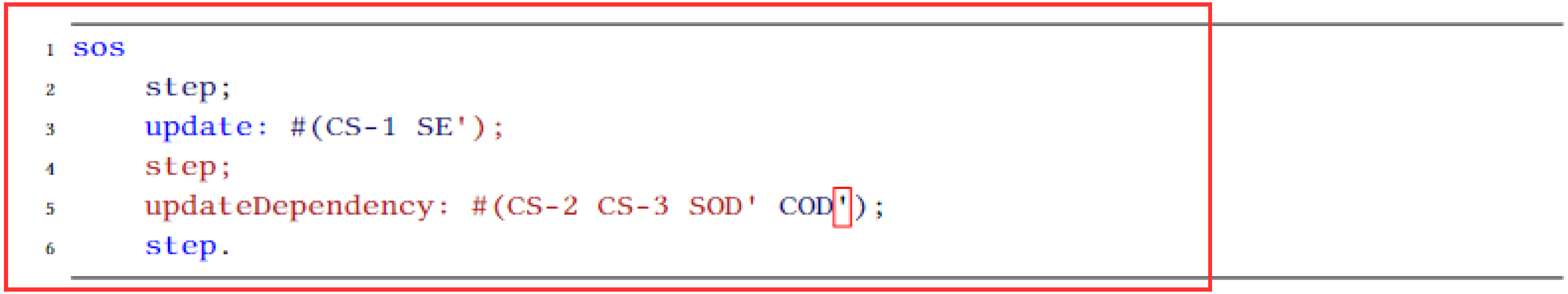


```
sos
    step;
    update: #(CS-1 SE');
    step;
    updateDependency: #(CS-2 CS-3 SOD' COD');
    step.
```

SoSAF: SM2 instance with Pharo language

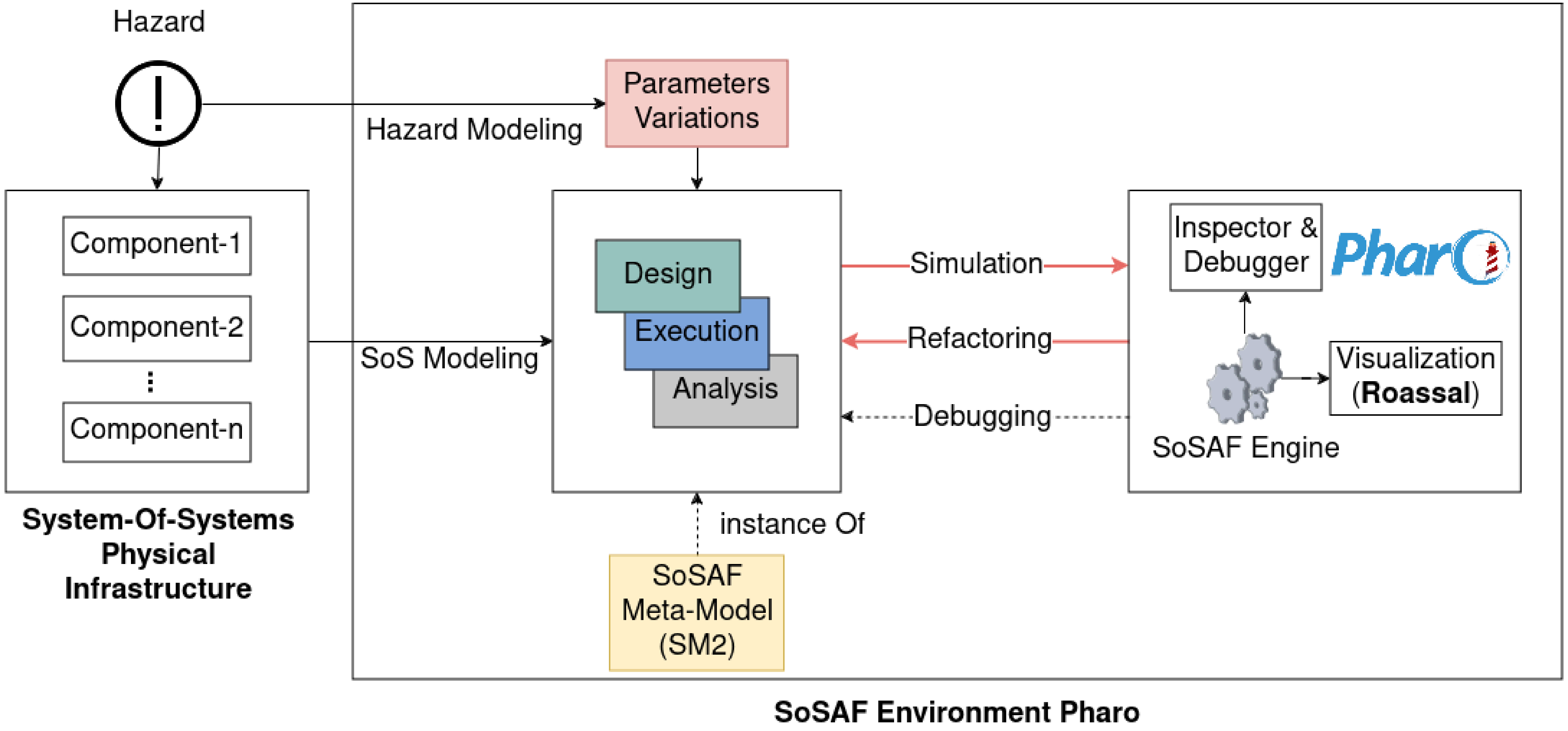


```
sos
        createCluster: #('cluster1' #(CS-1 CS-2));
        createCluster: #('cluster2' #(CS-3 CS-4)).
```



```
sos
    step;
    update: #(CS-1 SE');
    step;
    updateDependency: #(CS-2 CS-3 SOD' COD');
    step.
```

DEMO



Goals:

- Strengthen the *reliability* of a System of Systems
  - Develop intuitive and unified model for model designing
  - Develop Interactive Simulator
- Introduce failures through *dependency model* parameters

References

[1] **: Maier, Mark W**. "Architecting principles for systems-of-systems."
Systems Engineering: The Journal of the International Council on Systems Engineering 1.4 (1998)

[2] : **Xing, Liudong.** "Cascading failures in internet of things: review and perspectives on reliability and resilience."
IEEE Internet of Things Journal 8.1 (2020)

[3] : **Garvey, Paul R., and C. Ariel Pinto.** "Introduction to functional dependency network analysis."
The MITRE Corporation and Old Dominion, Second International Symposium on Engineering Systems, MIT, Cambridge, Massachusetts. Vol. 5. (2009)

[4] : **Guariniello, Cesare, and Daniel DeLaurentis**. "Supporting design via the system operational dependency analysis methodology."
Research in Engineering Design 28 (2017)