Towards Context-Aware Propagators

Language Constructs for Context-Aware Adaptation Dependencies

Engineer Bainomugisha, Wolfgang De Meuter, Theo D'Hondt

Programming Technology Lab, Vrije Universiteit Brussel, Belgium

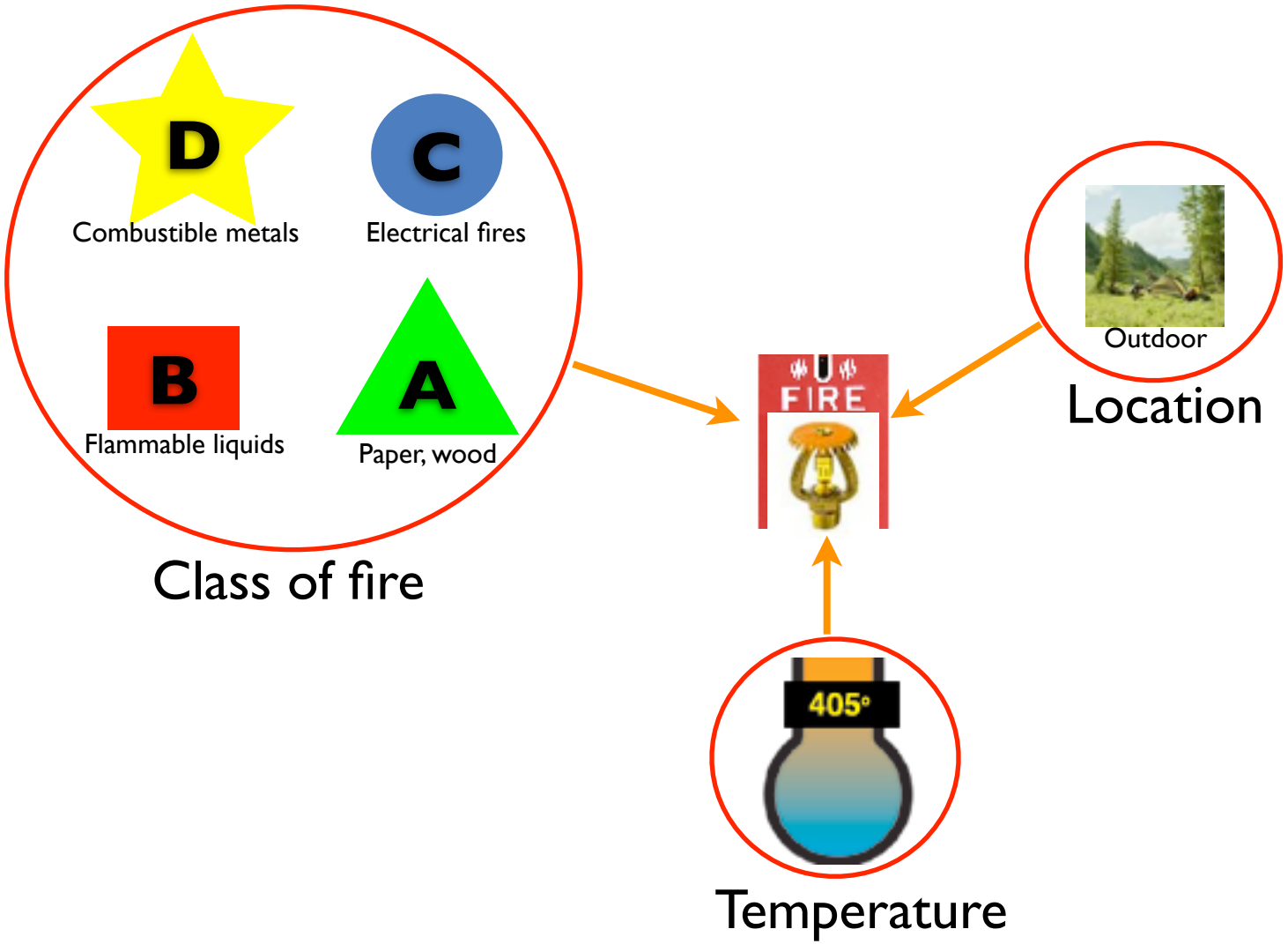
Goal

Ensuring consistency during system adaptation process

- ➡ Language support for selection of applicable adaptations
- ➡ (Re)definition of adaptation dependencies

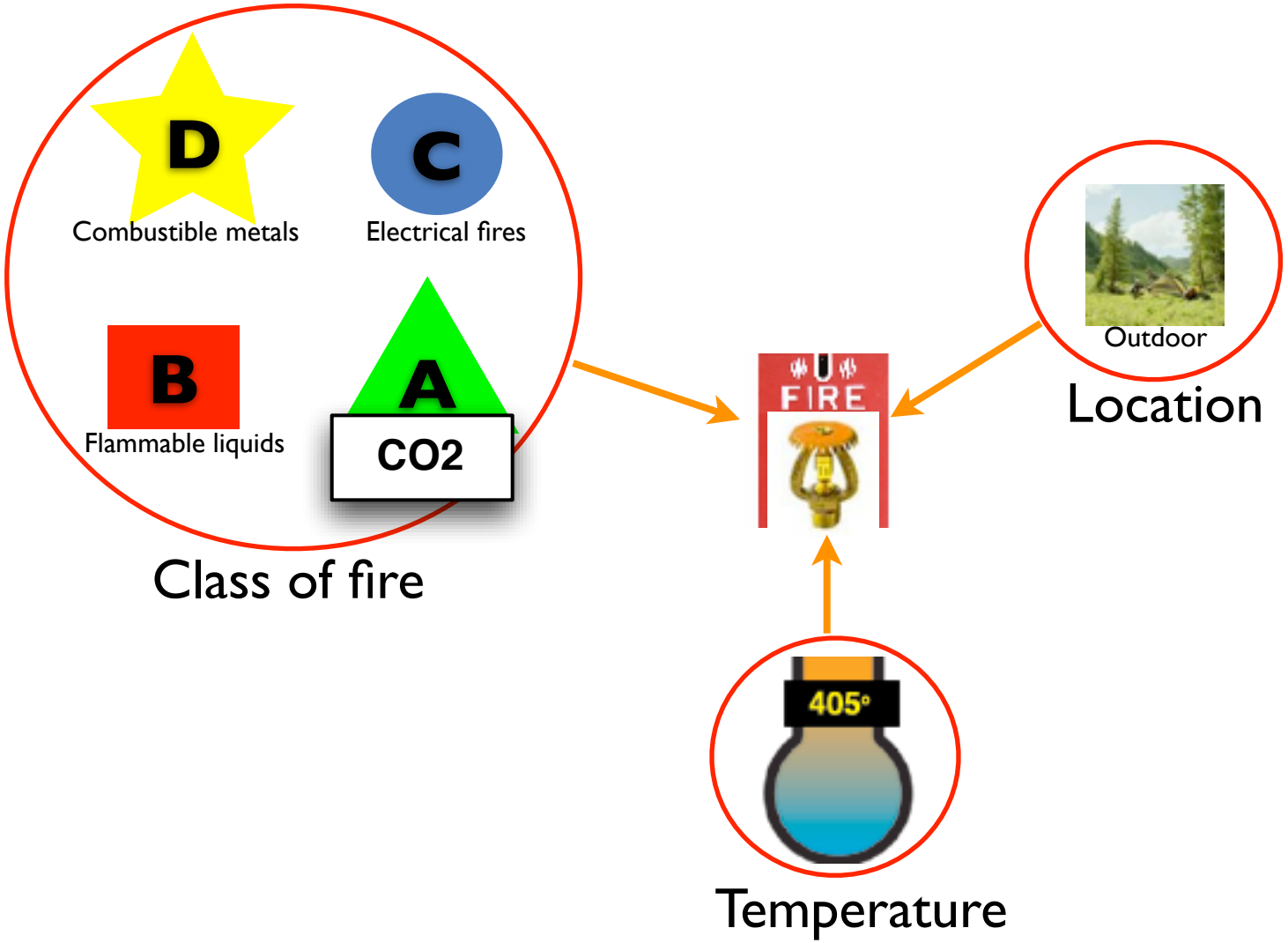
A Context-Aware System

Context-aware Fire Sprinkler System (CaFSS)



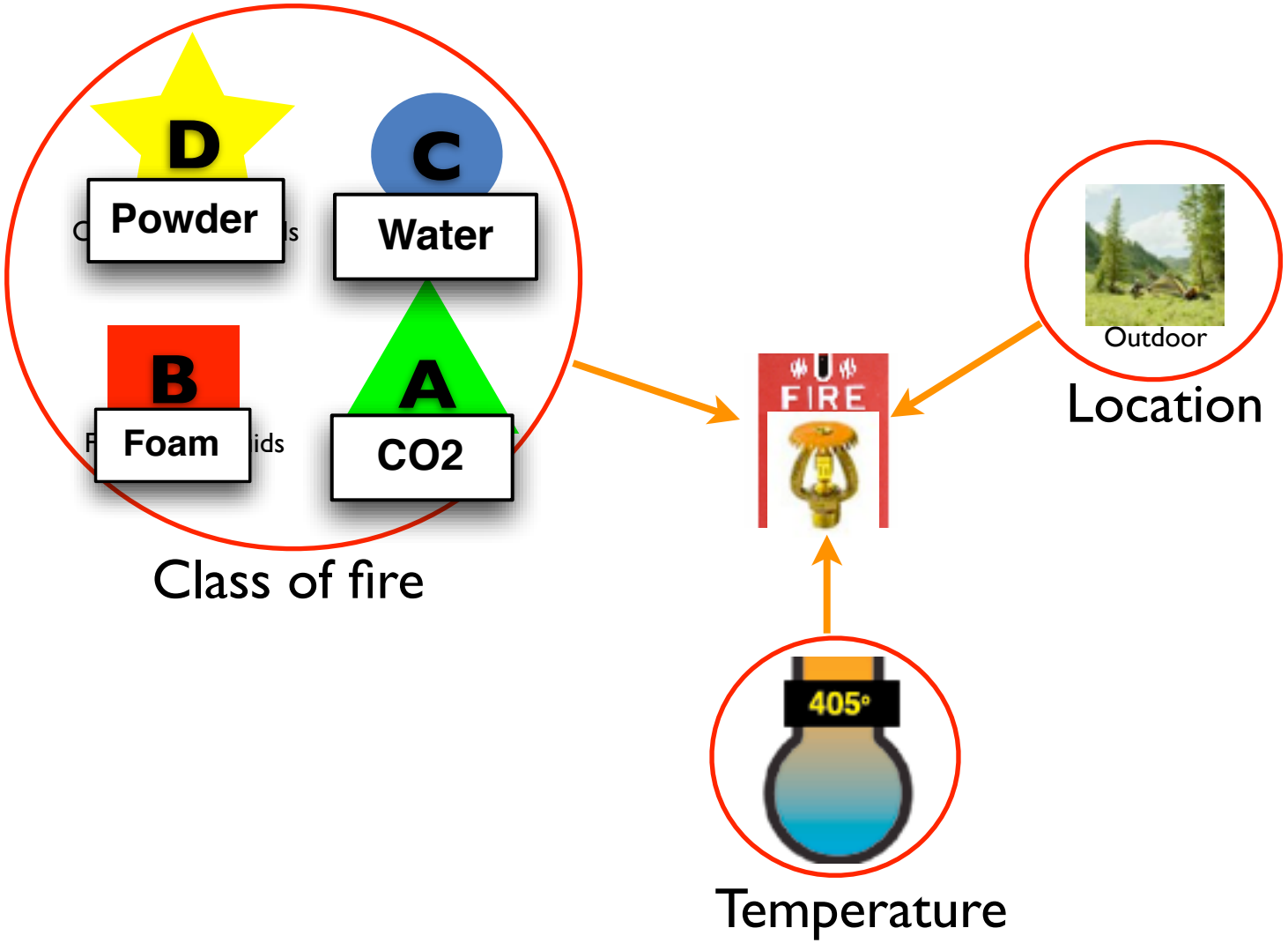
A Context-Aware System

Context-aware Fire Sprinkler System (CaFSS)



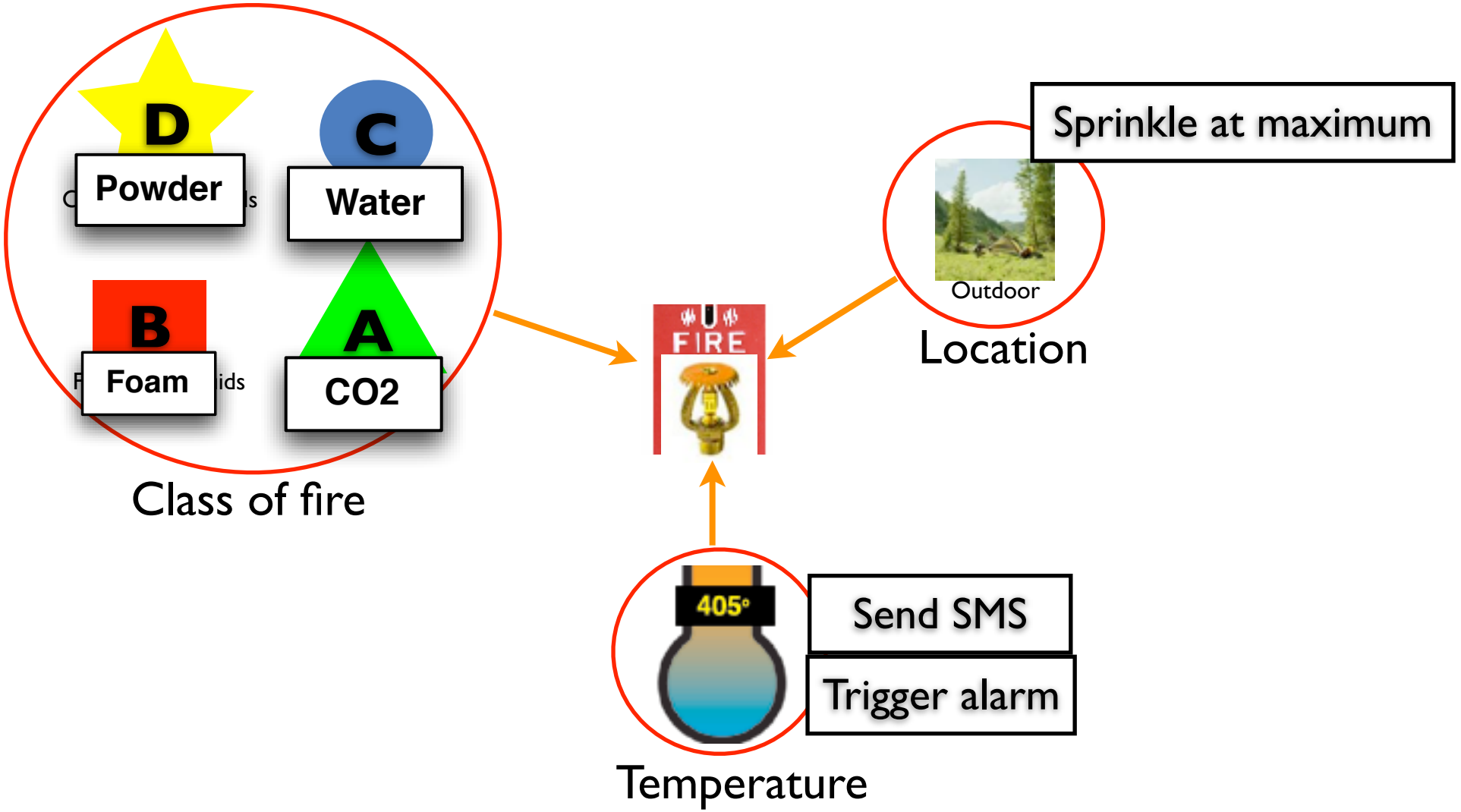
A Context-Aware System

Context-aware Fire Sprinkler System (CaFSS)



A Context-Aware System

Context-aware Fire Sprinkler System (CaFSS)

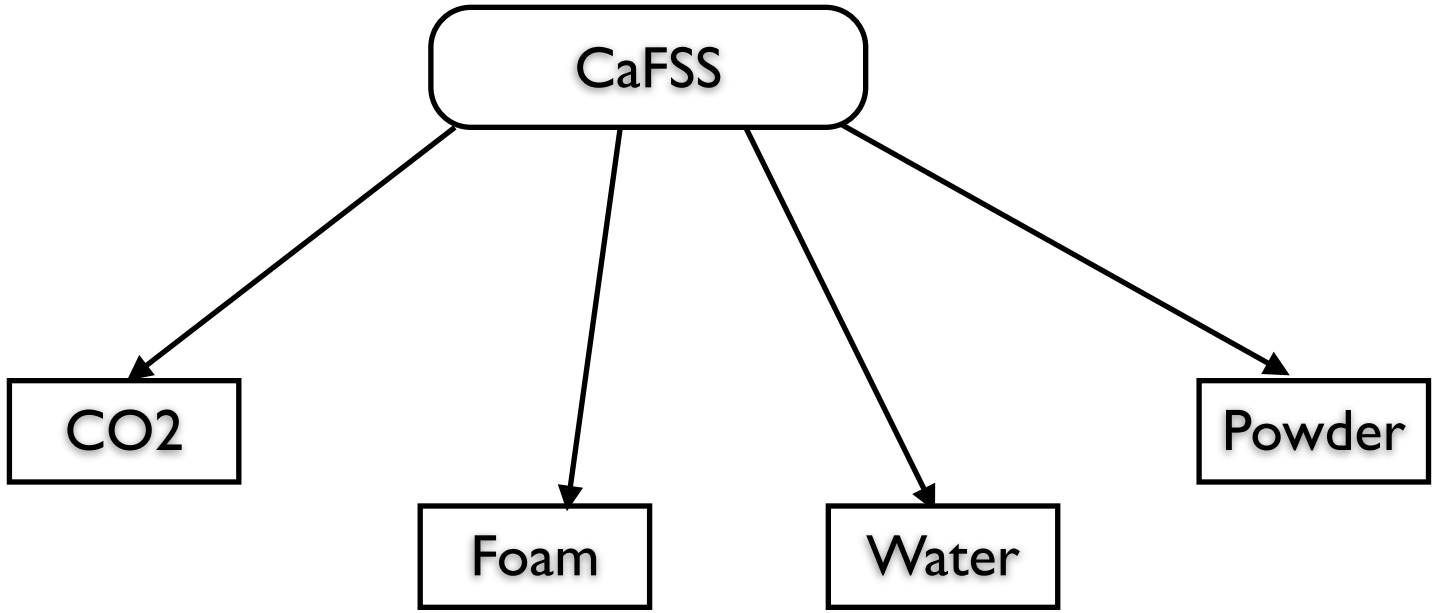


Existing Work

- Context-Oriented Programming (COP) *(Hirschfeld et.al.)*
  - Programming language support

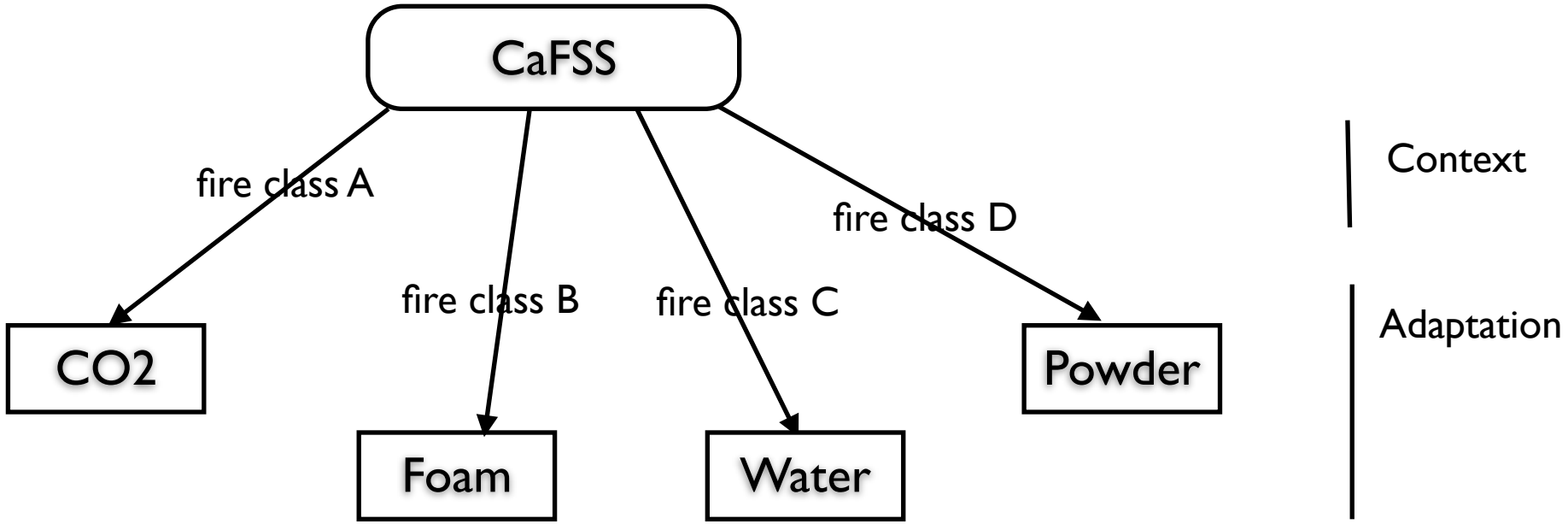
- Context-Oriented Domain Analysis (CODA) *(Desmet et.al.)*
  - High-level modelling approach for context-aware systems

CaFSS using CODA

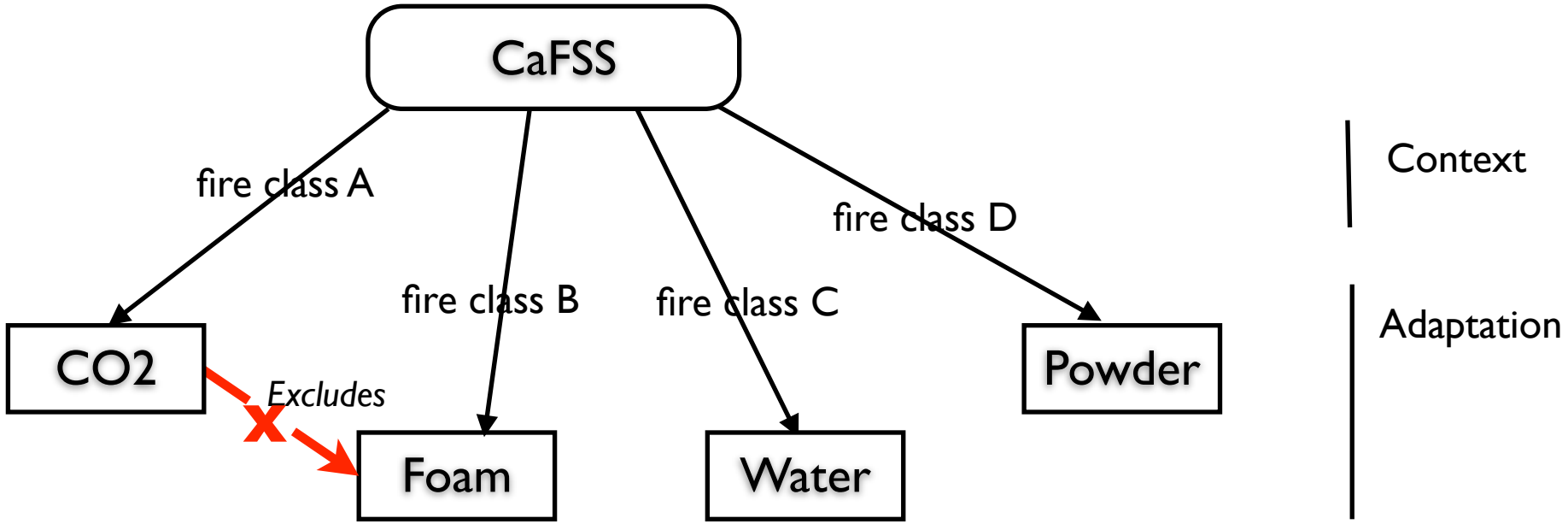


Adaptation

CaFSS using CODA

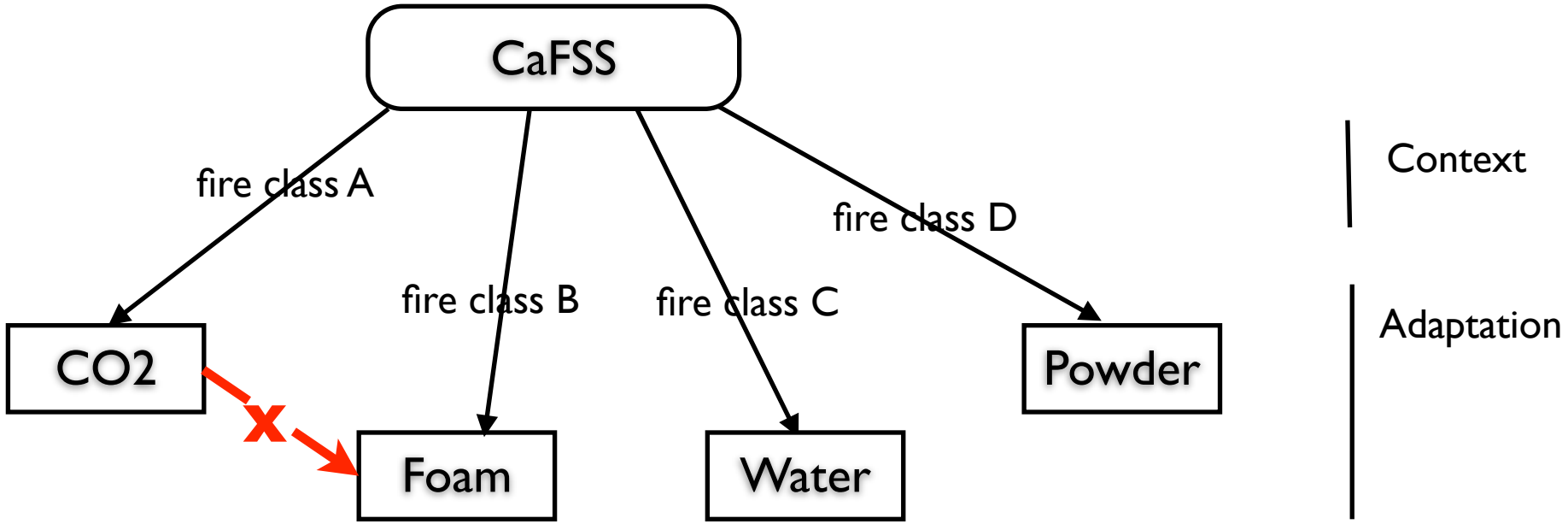


CaFSS using CODA



Adaptation dependencies

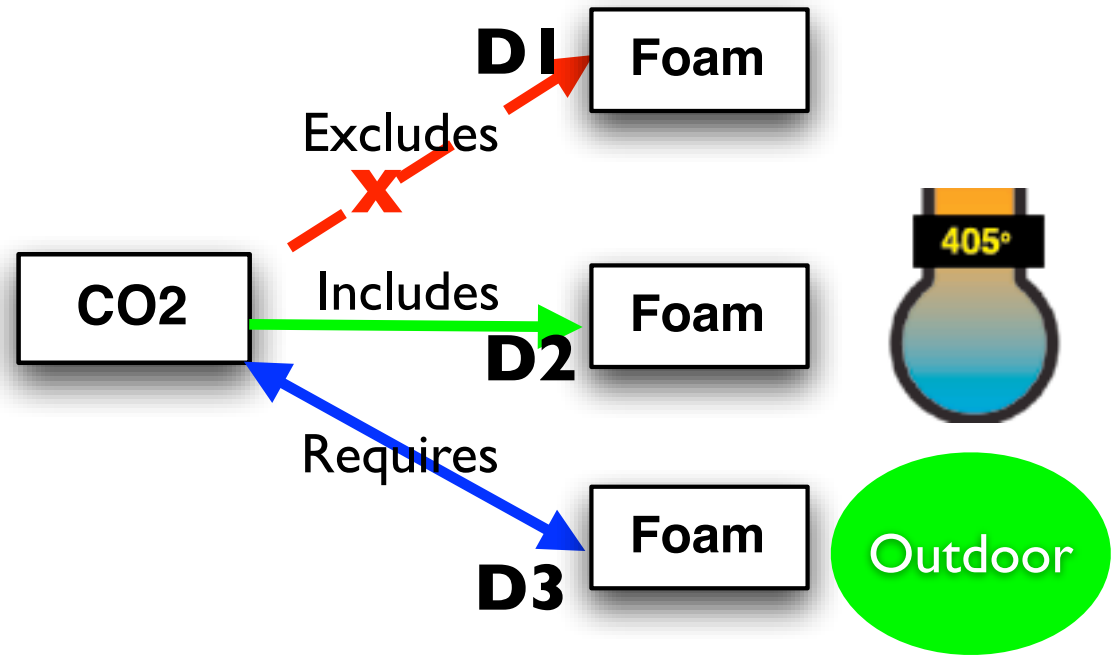
CaFSS using CODA



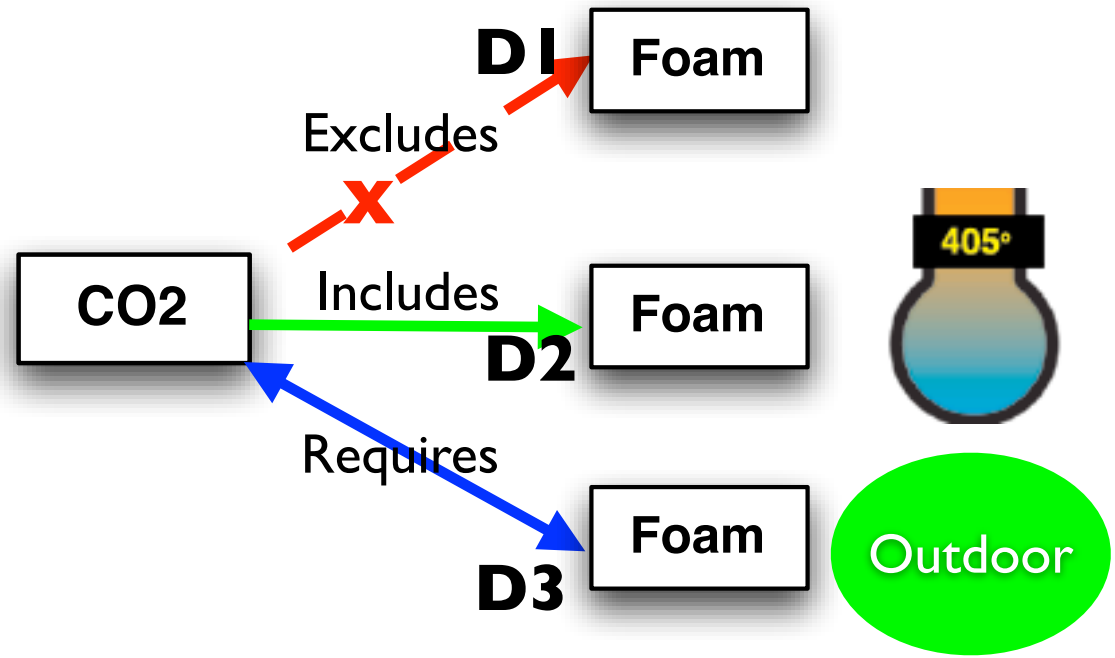
Issues

- Fixed dependencies e.g CO2 always excludes Foam
- Expressing all possibilities becomes cumbersome

Context-Aware Dependencies



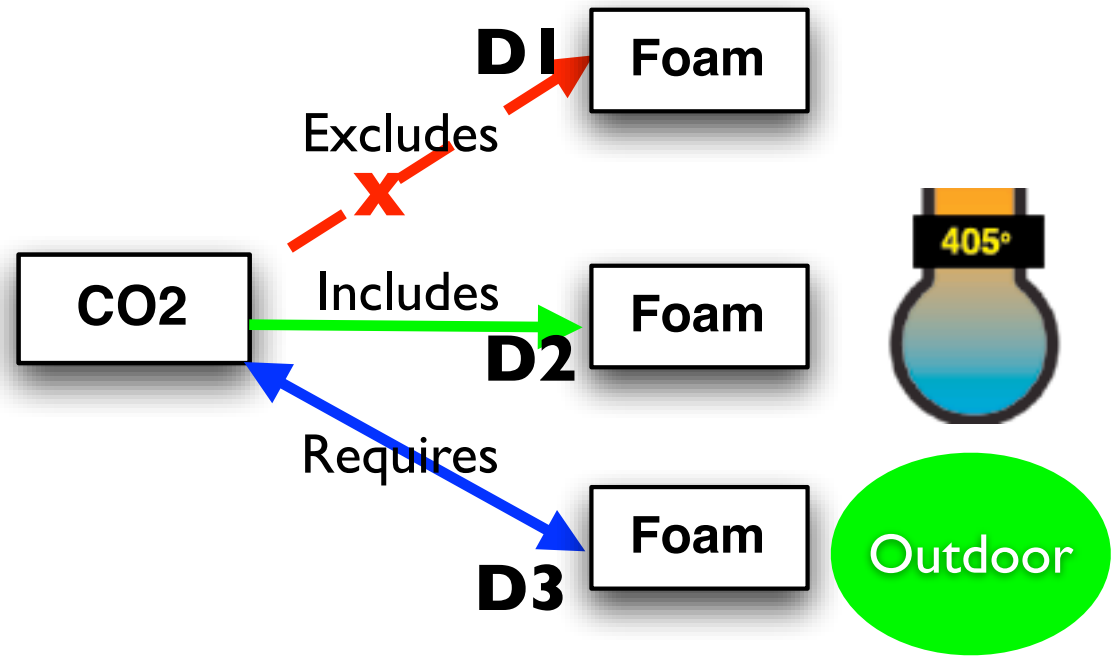
Context-Aware Dependencies



- Multiple dependencies can coexist

D := (D1 D2 D3)

Context-Aware Dependencies



- Multiple dependencies can coexist

D := (D1 D2 D3)

Contradictions

System Adaptation Issues

Need for Explicit Support for:

System Adaptation Issues

Need for Explicit Support for:

- Expression of context and adaptations

System Adaptation Issues

Need for Explicit Support for:

- Expression of context and adaptations
- (Re)Definition of adaptation dependencies

System Adaptation Issues

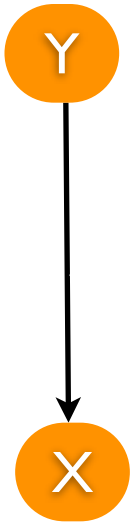
Need for Explicit Support for:

- Expression of context and adaptations
- (Re)Definition of adaptation dependencies
- Management of dependencies' contradictions

Context-Aware Propagators

Propagators model (*Radul, & Sussman, 2009*)

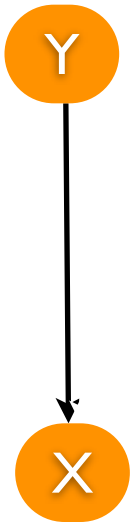
- Conventional programming



- One source for a variable's value

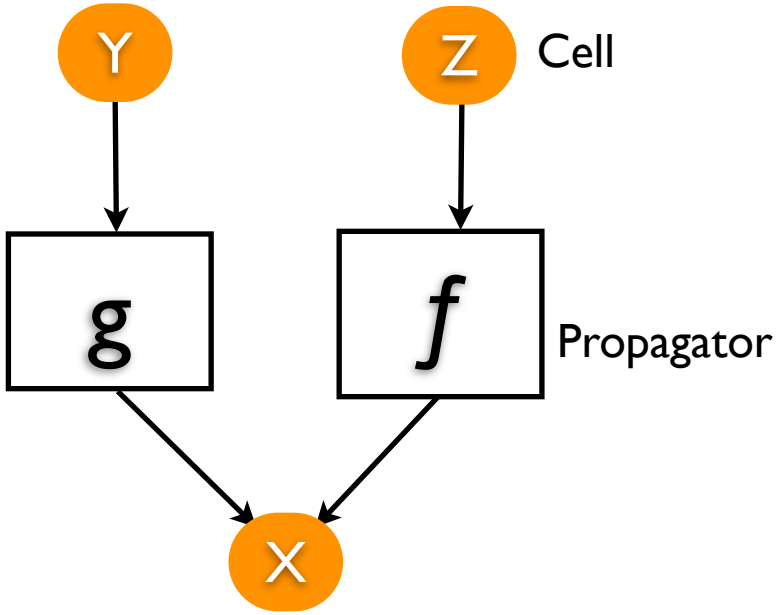
Context-Aware Propagators

- Conventional programming



- One source for a variable's value

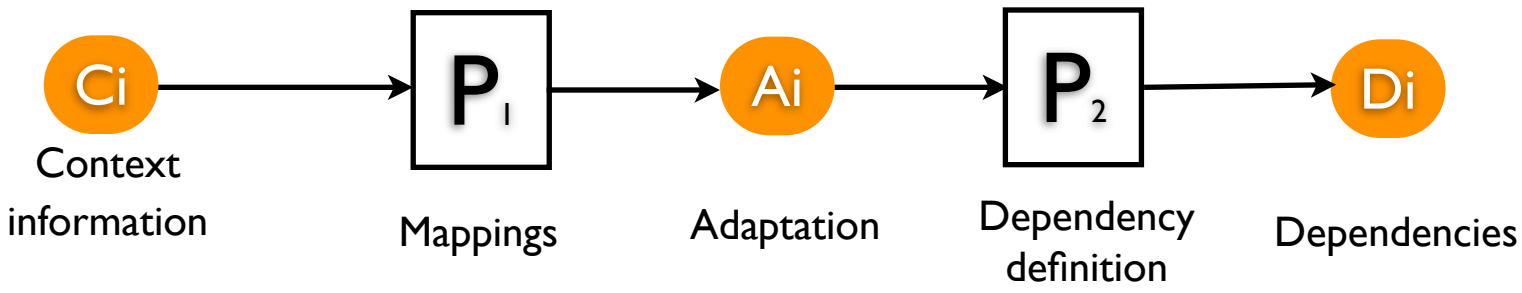
- Propagators model
(*Radul, & Sussman, 2009*)



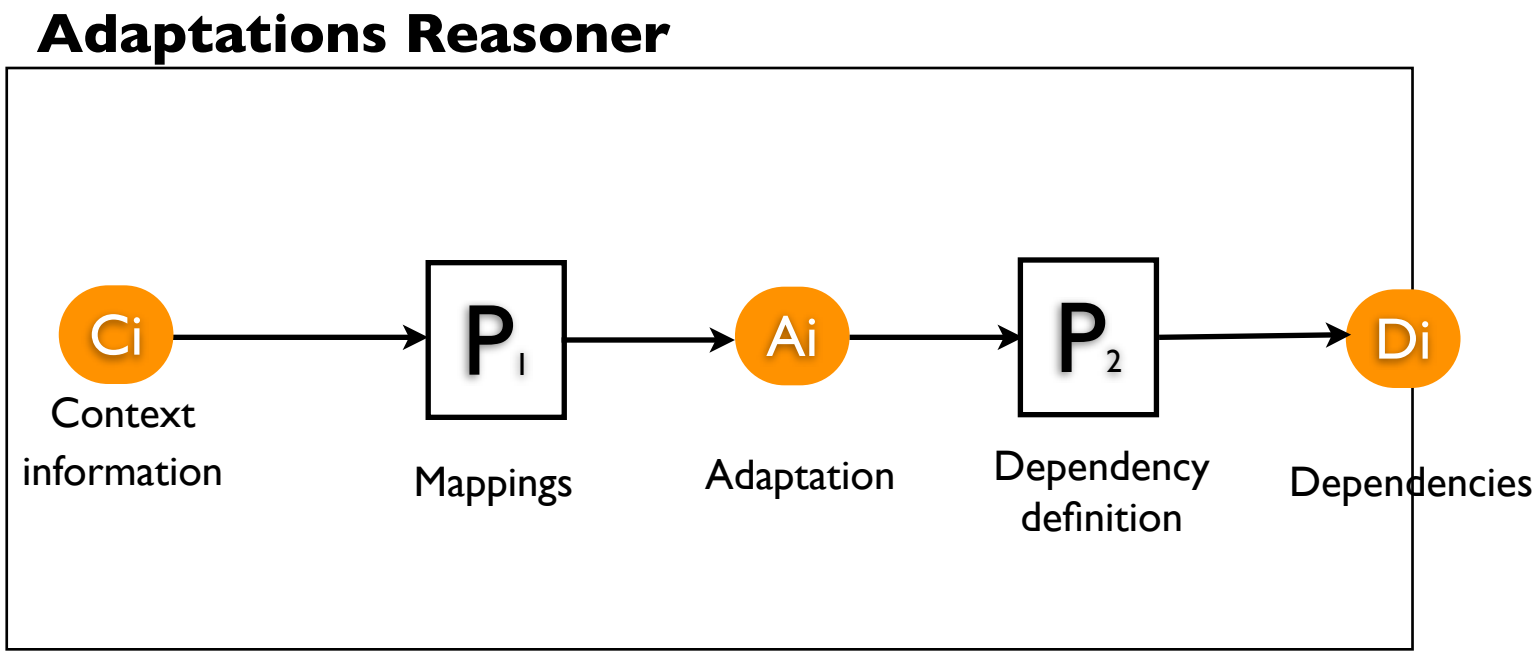
+ Multiple sources for the value

+ Coexisting values for a variable

Context-Aware Propagators



Context-Aware Propagators



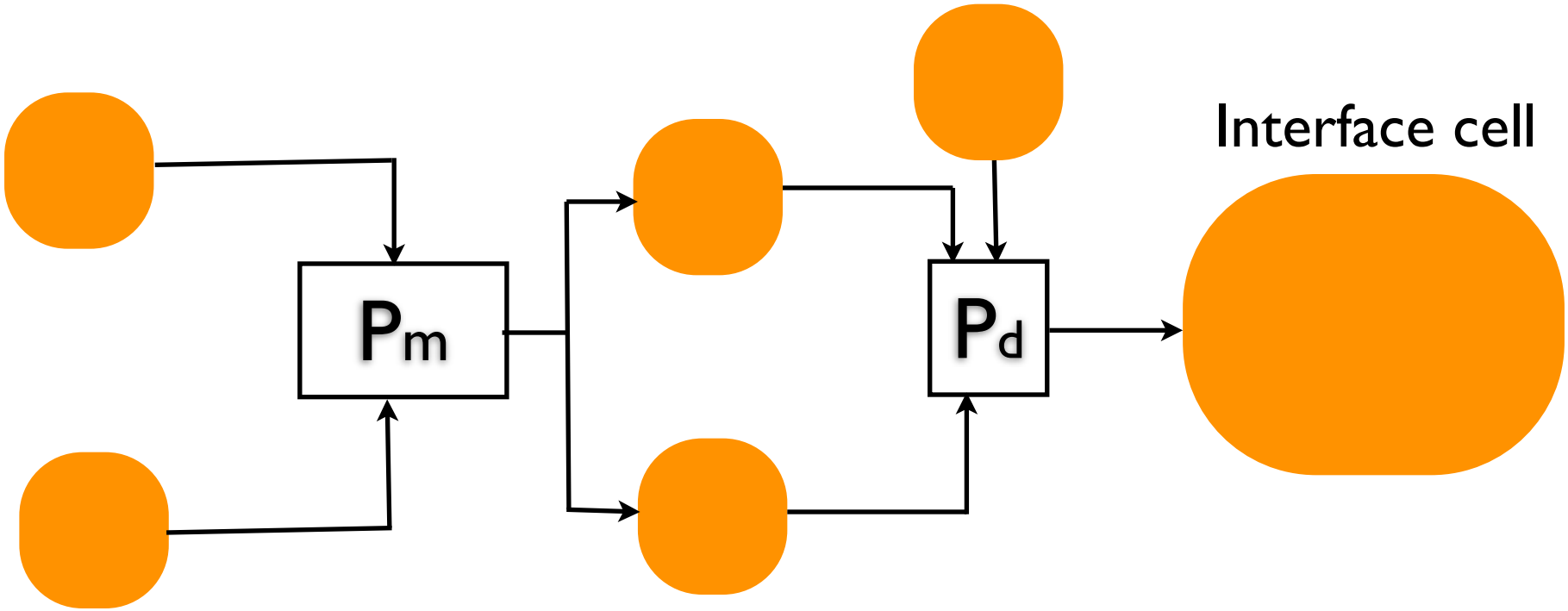
CaFSS using Propagators

```
(define kb (make-cell))
(define class-of-fire (make-cell))
(define temp (make-cell))
(define location (make-cell))
(define dependencies (make-interface-cell))
(CaFSS-reasoner
       kb class-of-fire temp location dependencies)

(content dependencies)
;;;Cell Value >>No Adaptation available

(add-content kb  *context-adaptation-kb*)
(add-content class-of-fire 'Class-B-Fire)
(content dependencies)
;;;Cell Value: (CO2)
```

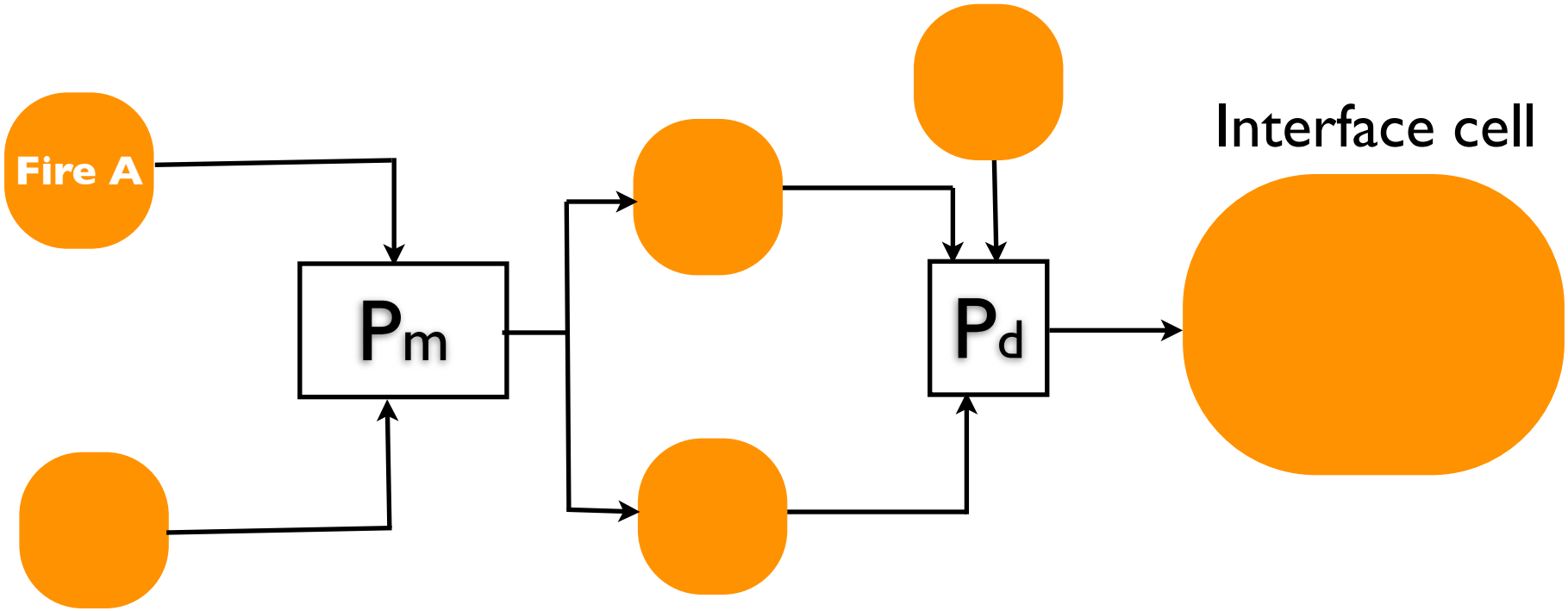
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

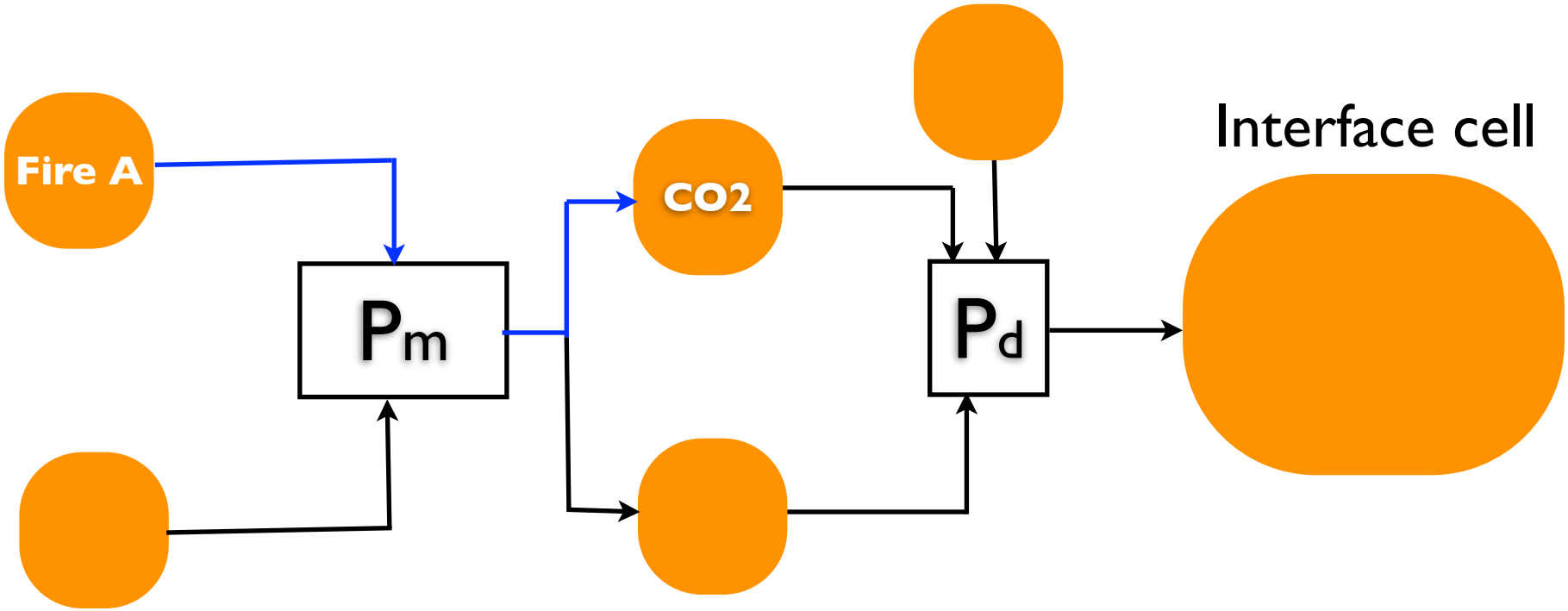
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

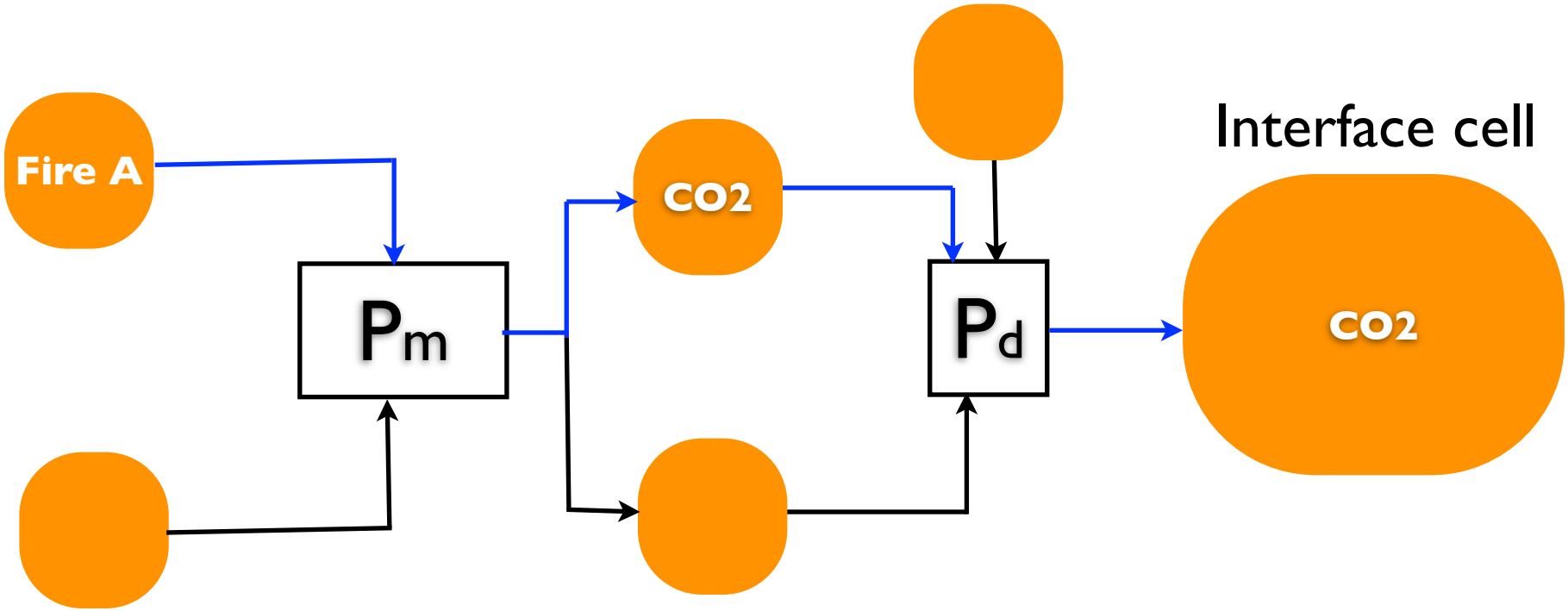
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

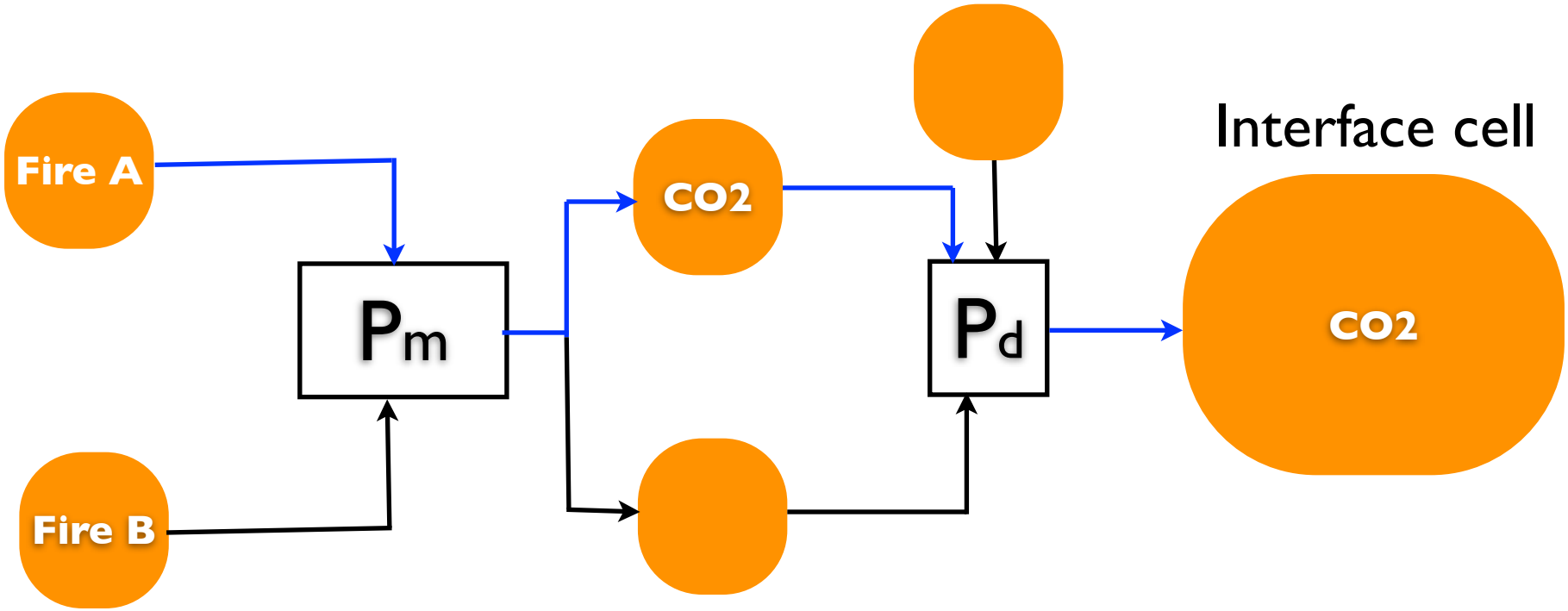
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

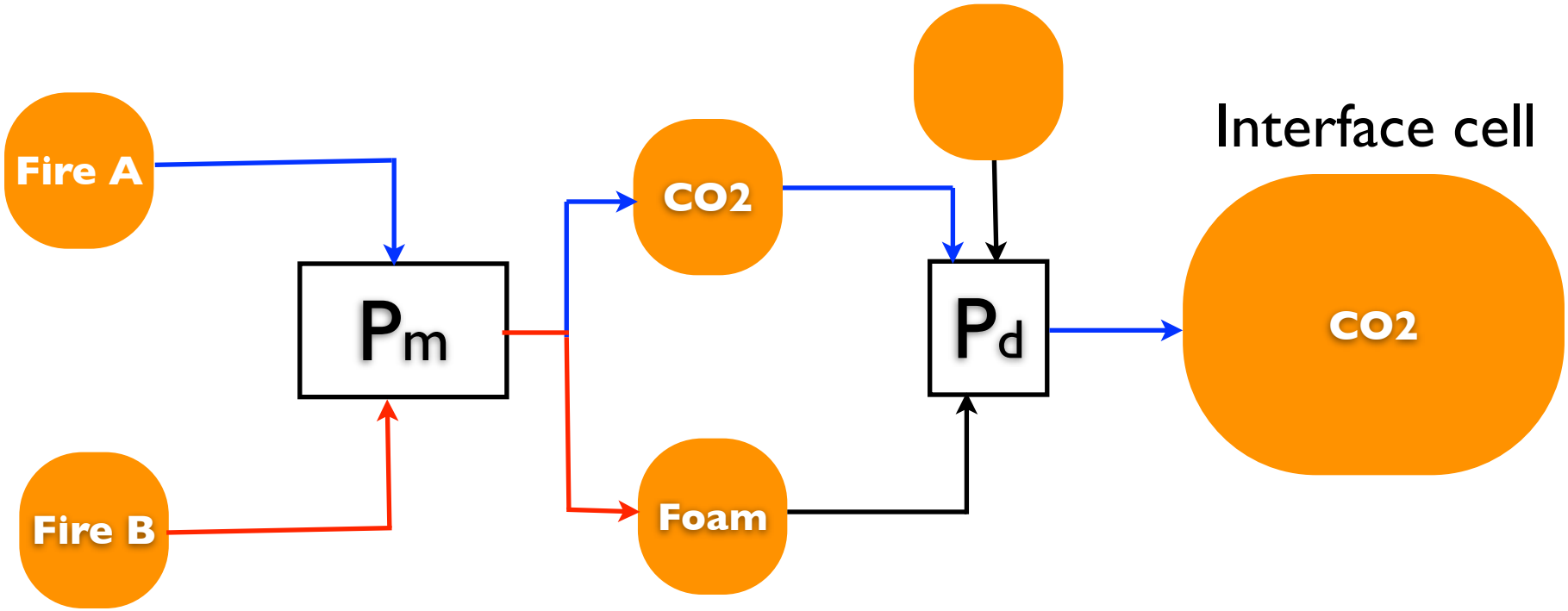
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

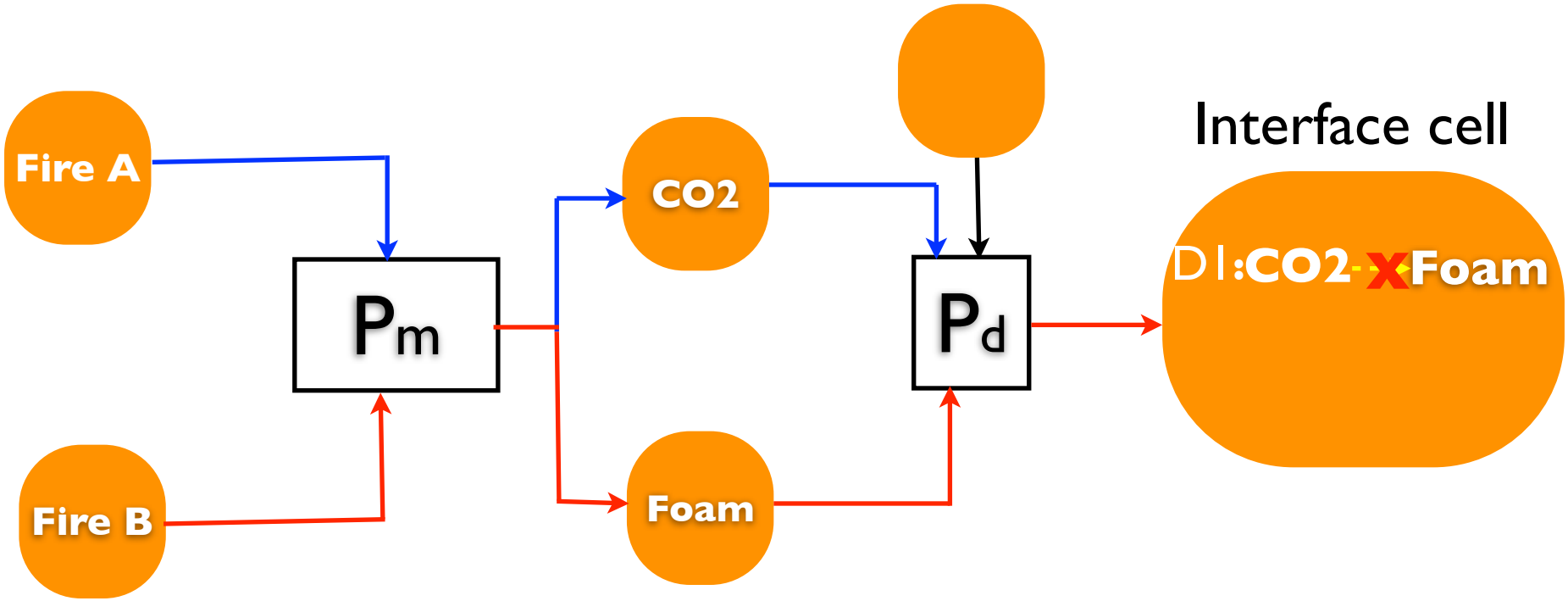
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

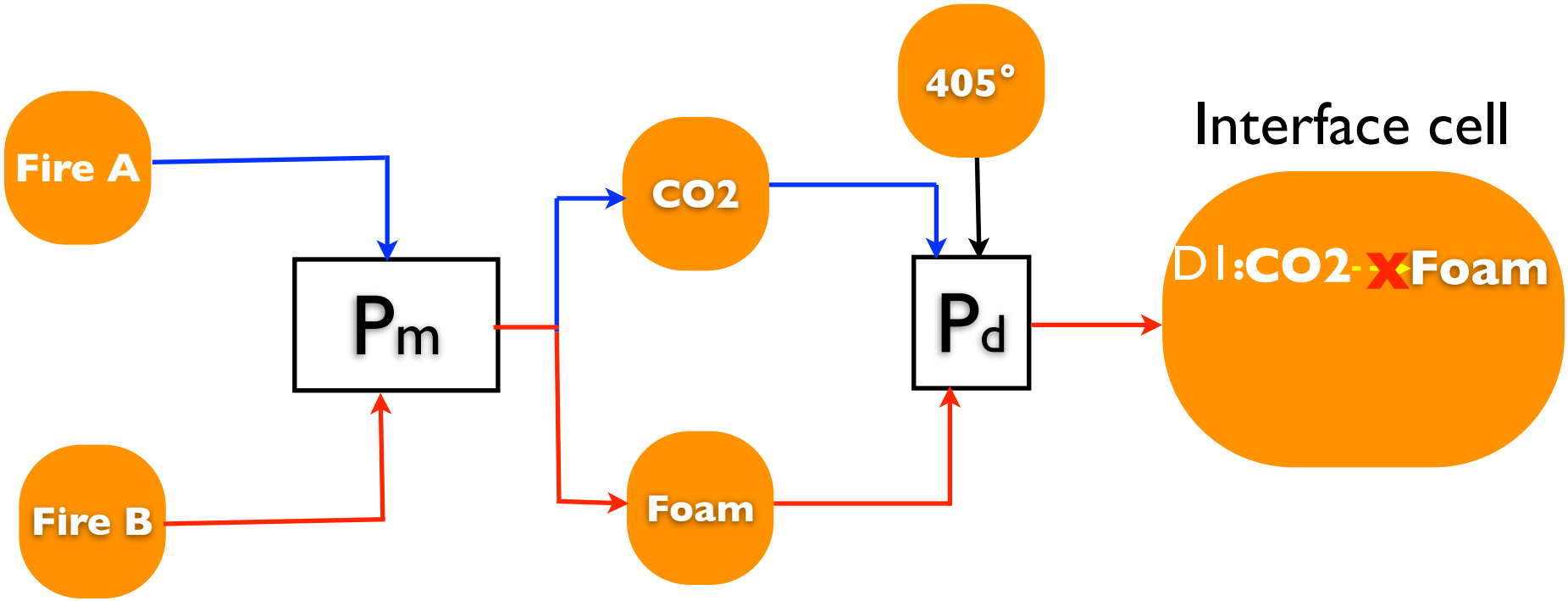
CaFSS using Propagators



Pm - Mappings propagator

Pd - Dependencies propagator

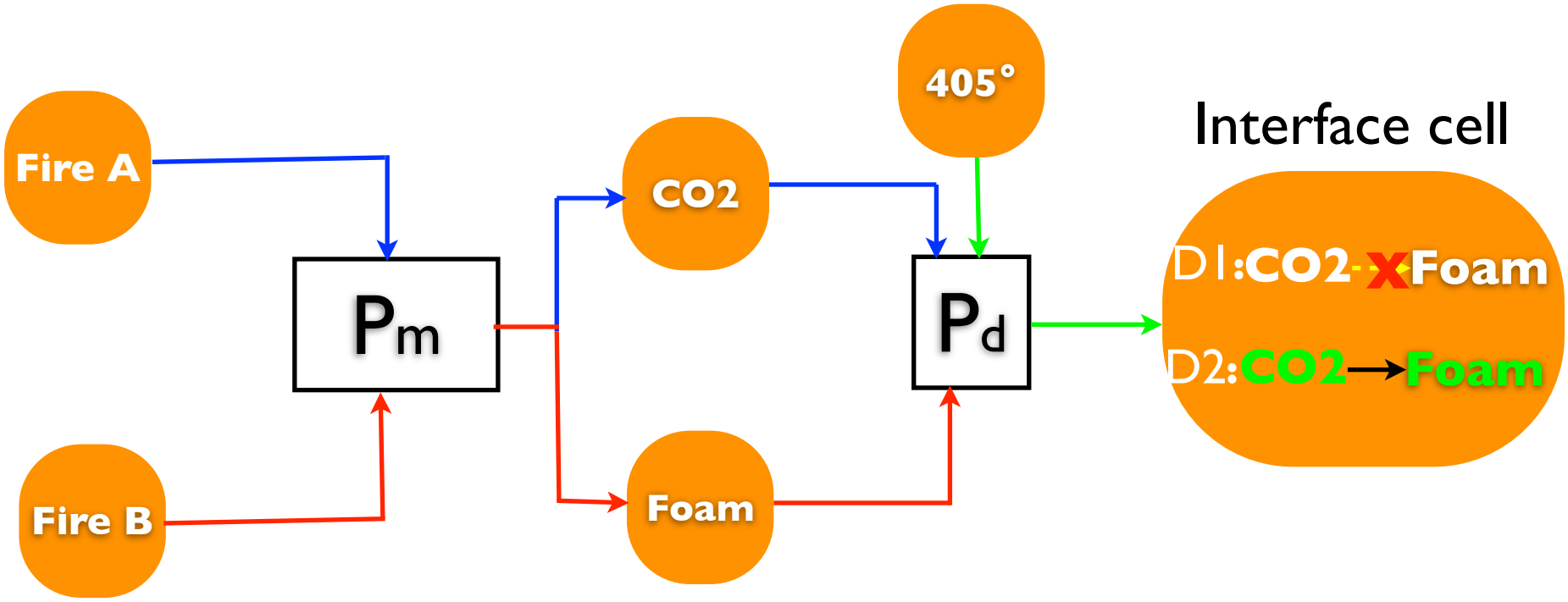
CaFSS using Propagators



Pm - Mappings propagator

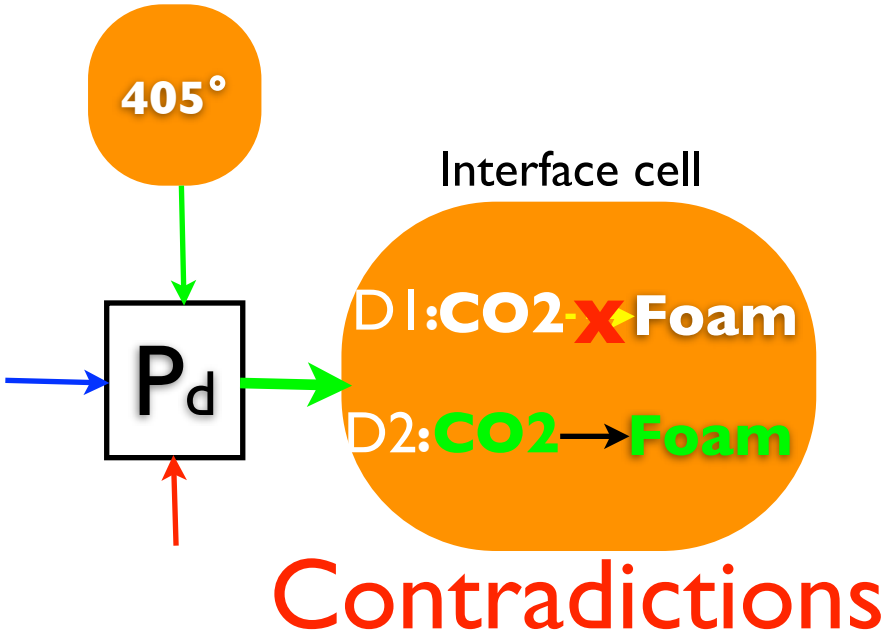
Pd - Dependencies propagator

CaFSS using Propagators



- Dependencies change depending on the context
- Multiple dependencies may contradict each other

Managing Dependencies' Contradictions



- Dependency "Decorations"
- Multidirectional Computation
- Merging

Summary

- Ensuring consistency through adaptation dependencies
- Context-aware dependencies: multiple dependencies may coexist
- Managing contradictions using propagators

Ongoing

- Further investigation on constraint propagation
- Implementation
- Propagators in a distributed setting
- Application to other scenarios

Thank you

Questions ?

ebainomu@vub.ac.be

http://prog2.vub.ac.be/~ebainomu/

References

- [2] B. Desmet, J. Vallejos, P. Costanza, andW. D. Meuter. Context-oriented domain analysis. *In proceedings of 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2007), Lecture Notes in Artificial Intelligence, Springer-Verlag*

- [4] R. Hirschfeld, P. Costanza, and O. Nierstrasz. Context-orientedprogramming. *Journal of Object Technology, 7(3):125–151–621, 2008.*

- [6] A. Radul andG. J. Sussman. The(abridged)art of the propagator. *In ILC2009: Proceedings of the International Lisp Conference2009. ACM, 2009.*