1-order

T

type

F

field

C

constructor

M

method

P

parameter

L

local variable

I_c

constructor call
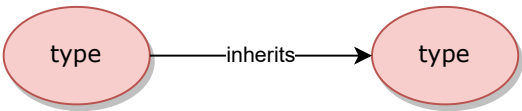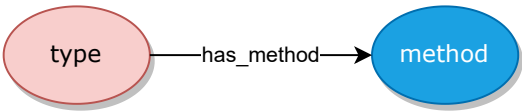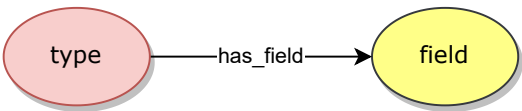
I_m

method call

2-order

TT

> *MP − Taxonomy*
> *DPC − Object structure child*
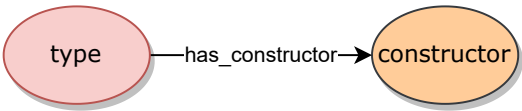> *EDP − Inheritance*

TM

> *MP − Function Pointer, Data Manager, State Machine, Pure Type*
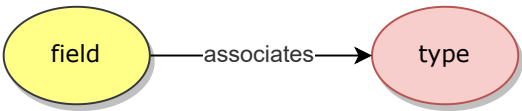> *EDP − Abstract Interface*

TF

> *MP − Pool, Stateless, Common State, Immutable, Box, Compound Box, Canopy, Record, Trait*
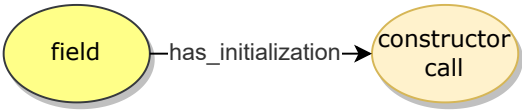> *DPC − Private flag, Static flag*

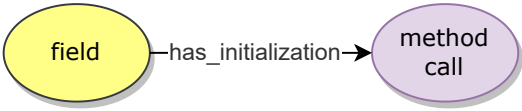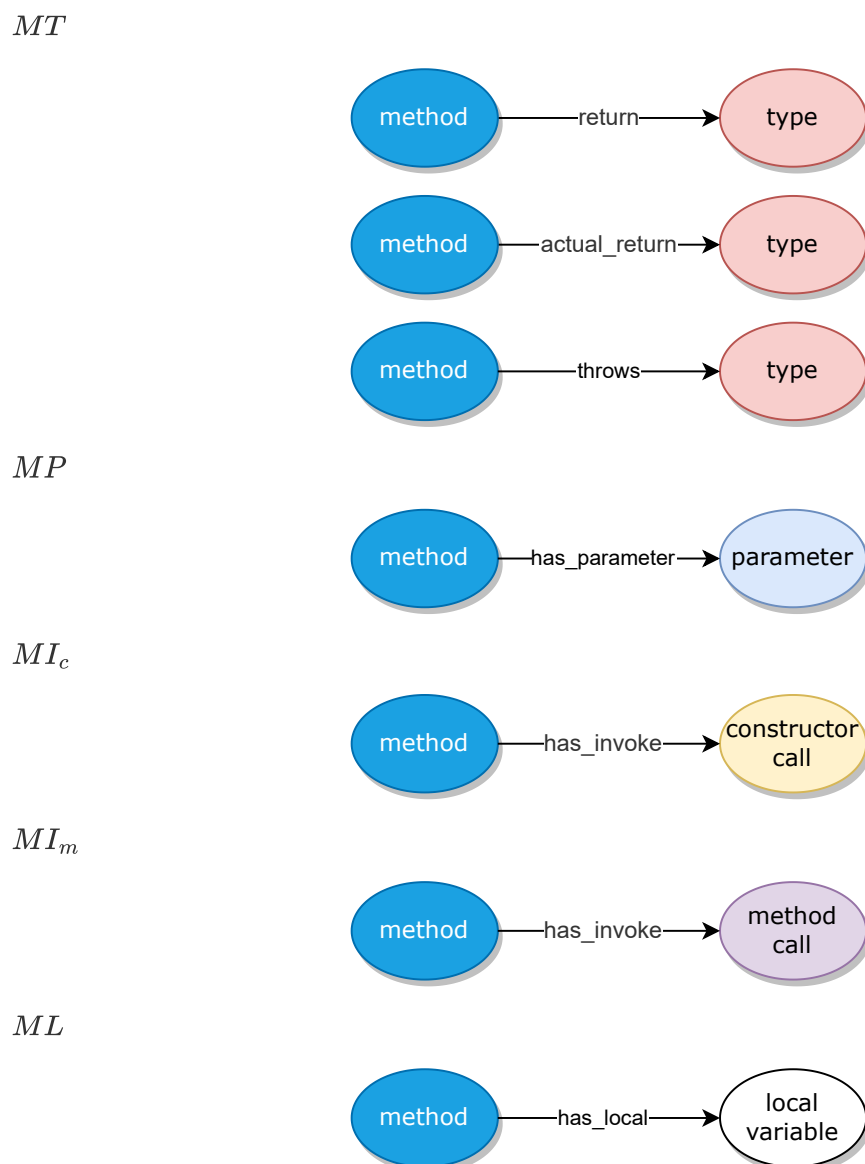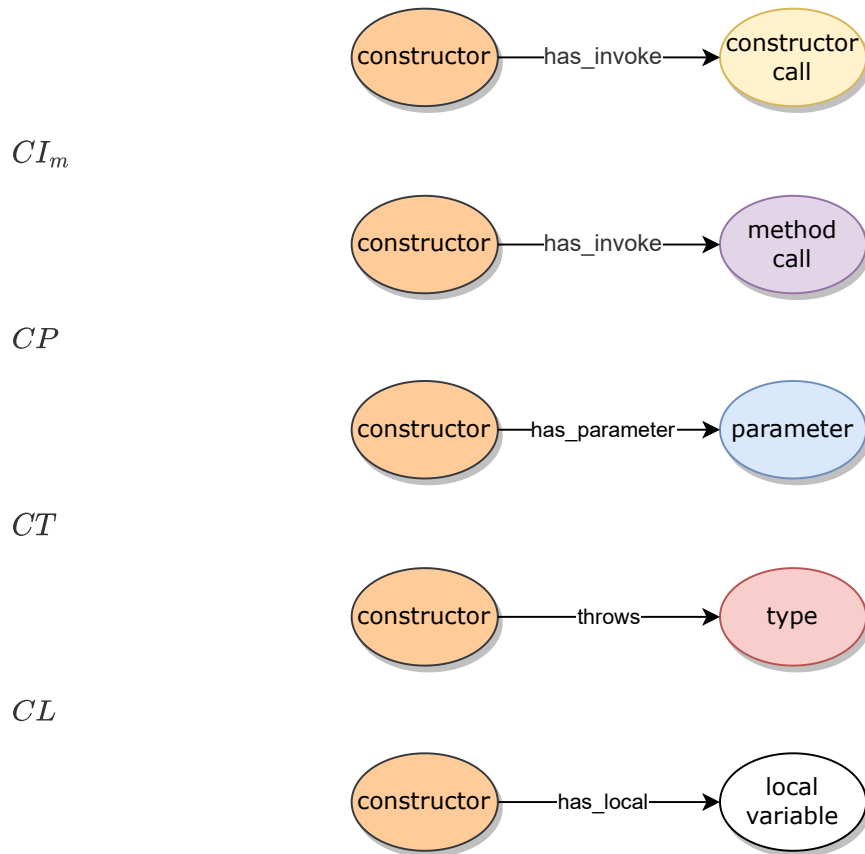*TC*

> *DPC − Protected instantiation*

FT

FI_c

FI_m

CI_c

CI_m

constructor —has_invoke→ constructor call

CP

constructor —has_invoke→ method call

CT

constructor —has_parameter→ parameter

CL

constructor —throws→ type

constructor —has_local→ local variable

MT

method —return→ type

method —actual_return→ type

method —throws→ type

MP

method —has_parameter→ parameter

MI_c

method —has_invoke→ constructor call

MI_m

method —has_invoke→ method call

ML

method —has_local→ local variable

MM

method —overrides/implements→ method

PT

parameter —associaties→ type

LT

local variable —associaties→ type

LI_c

local variable —has_initialization→ constructor call

LI_m

local variable —has_initialization→ method call

I_mM

method call —target→ method

I_mI_m

method call —next→ method call

method call —has_ananoymous_argument→ method call

I_mI_c

method call —has_ananoymous_argument→ constructor call

I_mF

method call —caller→ field

I_mP

$I_m L$

method call —caller→ parameter

method call —caller→ local variable

$I_c C$

constructor call —target→ constructor

$I_c I_c$

constructor call —has_ananoymous_argument→ constructor call

$I_c I_m$

constructor call —next→ method call

constructor call —has_ananoymous_argument→ method call

3-order

TT^2

type —inherits→ type ←inherits— type

Simple Factory, Abstract Factory, Mediator

type —inherits→ type
type —inherits→ type
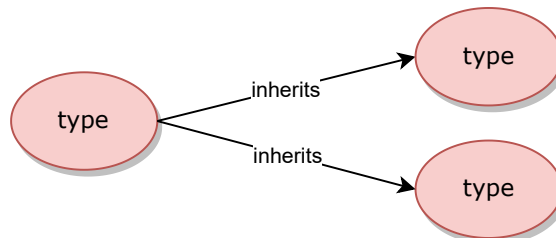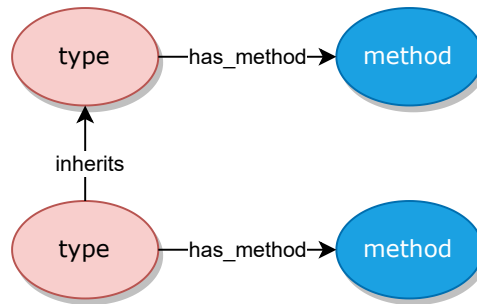
$MP - Joiner$
$DPC - Interface\ and\ class\ inherited,\ Multiple\ interfaces\ inherited$

TM^2
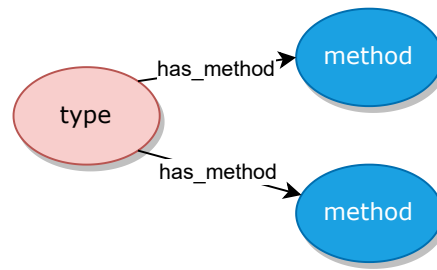
Memento

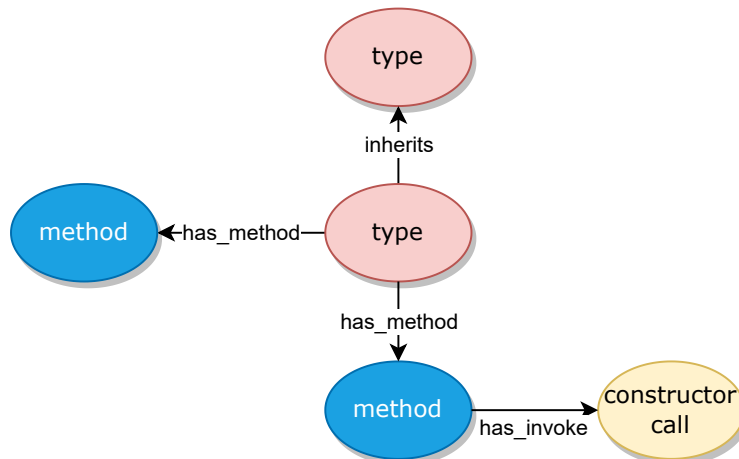$$TM \times 2 \cup TT$$

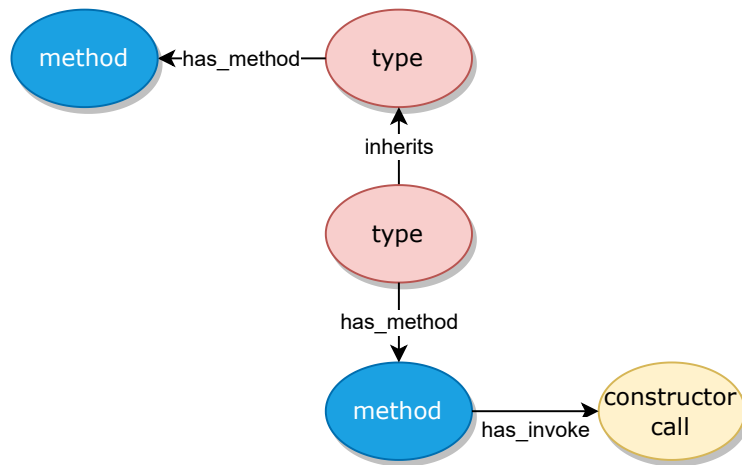$MP - Extender$

$$TMI_c$$

$DPC - Void\ return$
$EDP - Create\ Object$

$$TMI_c \cup TT \cup TM$$

TMI_m

TMP

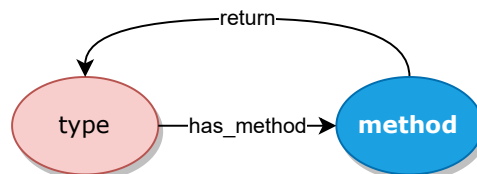*DPC − Controlled parameter*

TMT

DPC − Empty concrete product getter, Empty method, Cross hierarchy return

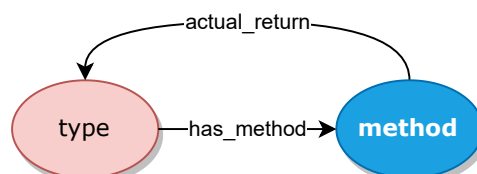Factory Method, Abstract Factory, Prototype, Memento
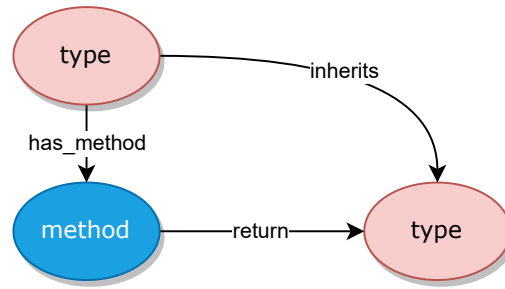

DPC − Concrete product getter

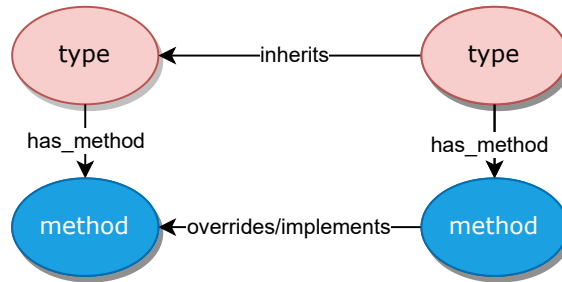Factory Method, Abstract Factory, Builder, Prototype

Singleton

$TMT \cup TT$

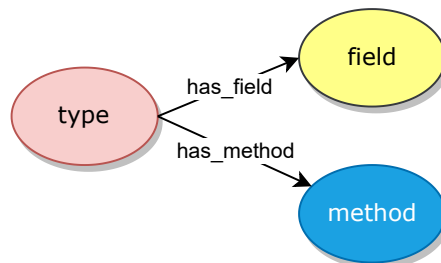*DPC − Parent Product Returned*

$TMM \cup TM \cup TT$

MP − Implementor, Overrider
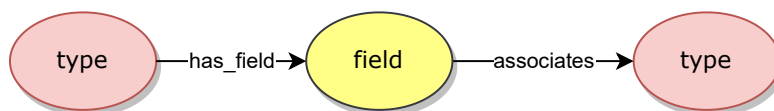*DPC − Interface method, Overriding method*

Factory Method, Abstract Factory, Builder, Prototype, Adapter(object), Bridge, Flyweight, Chain of Responsibility, Command, Iterator, Observer, State, Strategy, Template Method, Visitor
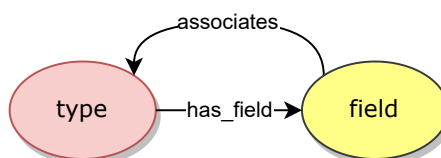
$TM \cup TF$

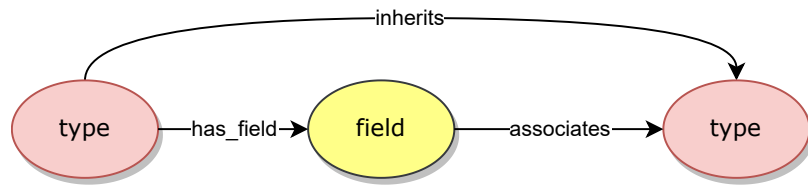*MP − Function Object, Cobol Like, Augmented Type, Pseudo Class*

TFT

DPC − Instance in abstract class

Builder, Prototype, Adapter(object), Bridge, Composite, Decorator, Proxy, Command, Iterator, Mediator, Memento, Observer, State, Strategy
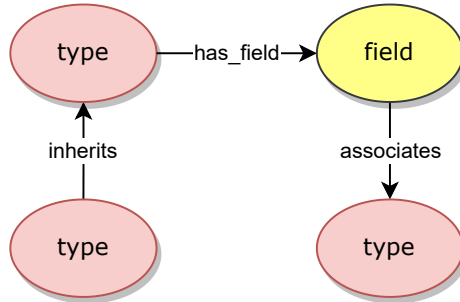

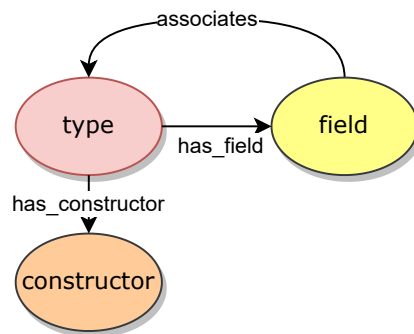*DPC − Private self instance, Static self instance, Single self instance*
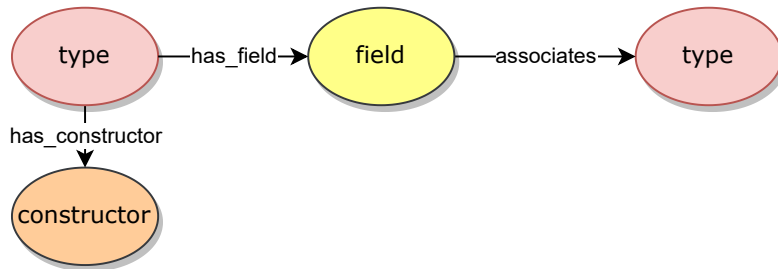
$TFT \cup TT$

$TFT \cup TC$

TFI_c

TCI_c

4-order
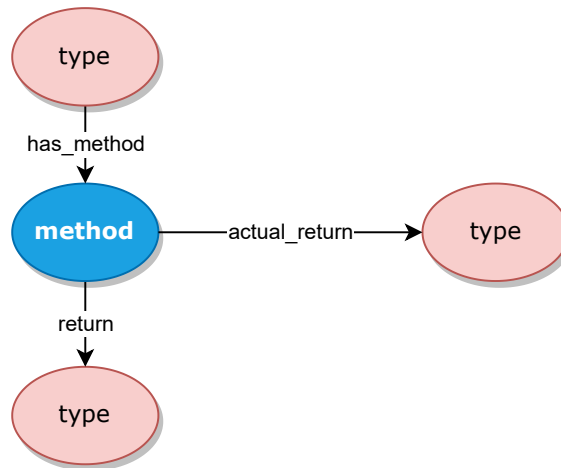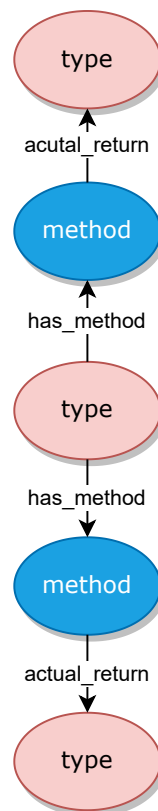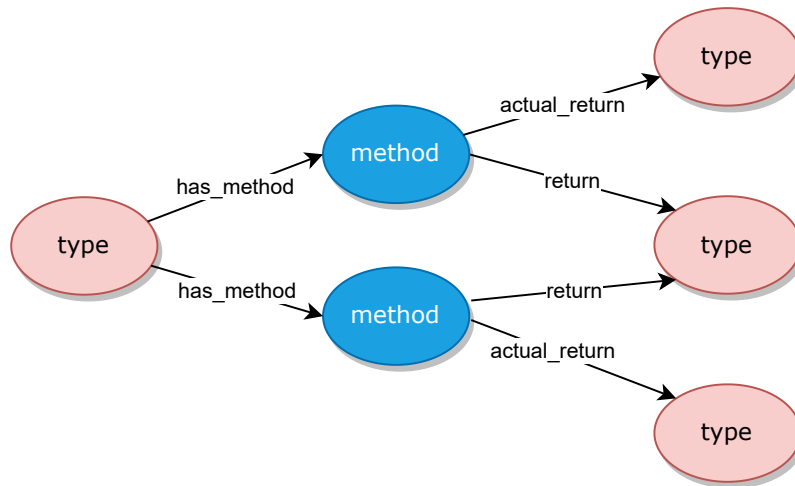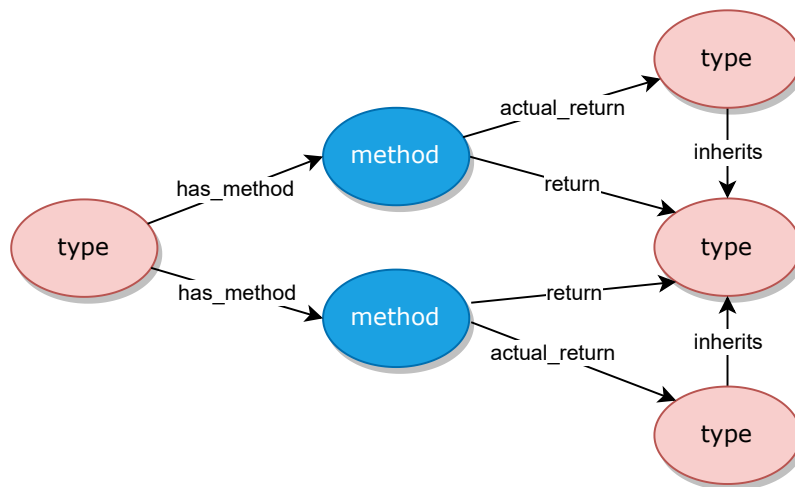
TMT^2

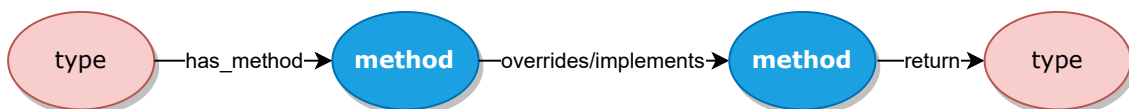$TMT^2 \cup TT$
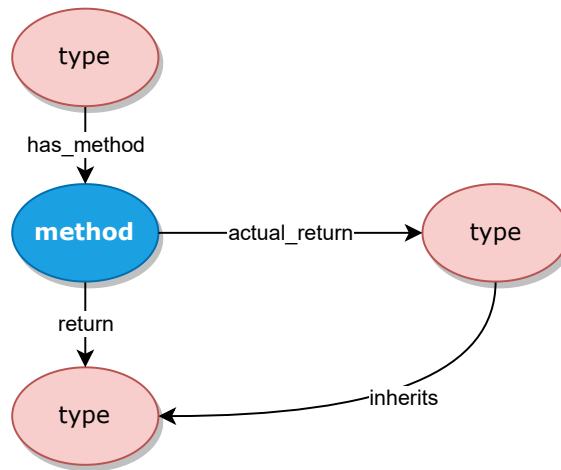
DPC − Concrete products returned

Simple Factory

TMT^4
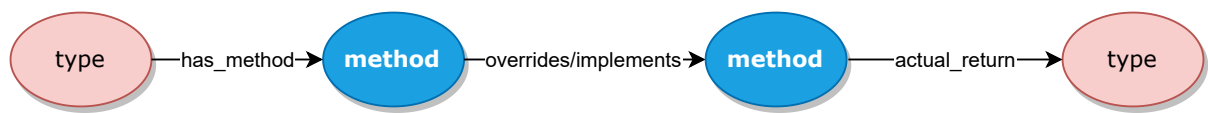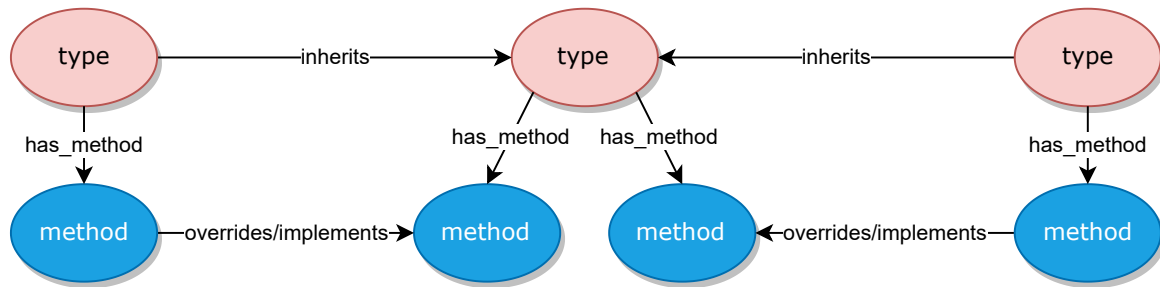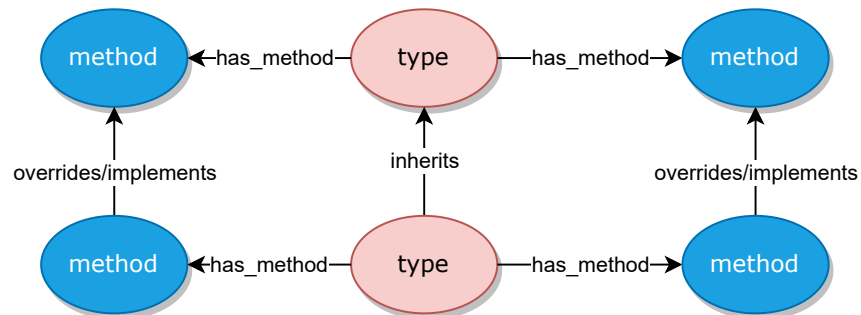

$TMT^4 \cup TT^2$

Simple Factory

$TMMT$

type —has_method→ **method** —overrides/implements→ **method** —actual_return→ type

$$(TMM \cup TM \cup TT)^2$$

type —inherits→ type ←inherits— type

has_method

type —has_method→ method ←has_method— type has_method

method —overrides/implements→ method   method ←overrides/implements— method

Factory Method

method ←has_method— type —has_method→ method

overrides/implements   inherits   overrides/implements

method ←has_method— type —has_method→ method

Builder, Template Method

method ←has_method— type   type —has_method→ method

overrides/implements   inherits   inherits   overrides/implements

type

has_method   has_method

method   method

Adapter(class)

type —inherits→ type ←inherits— type

has_method   has_method   has_method

method —overrides/implements→ method ←overrides/implements— method
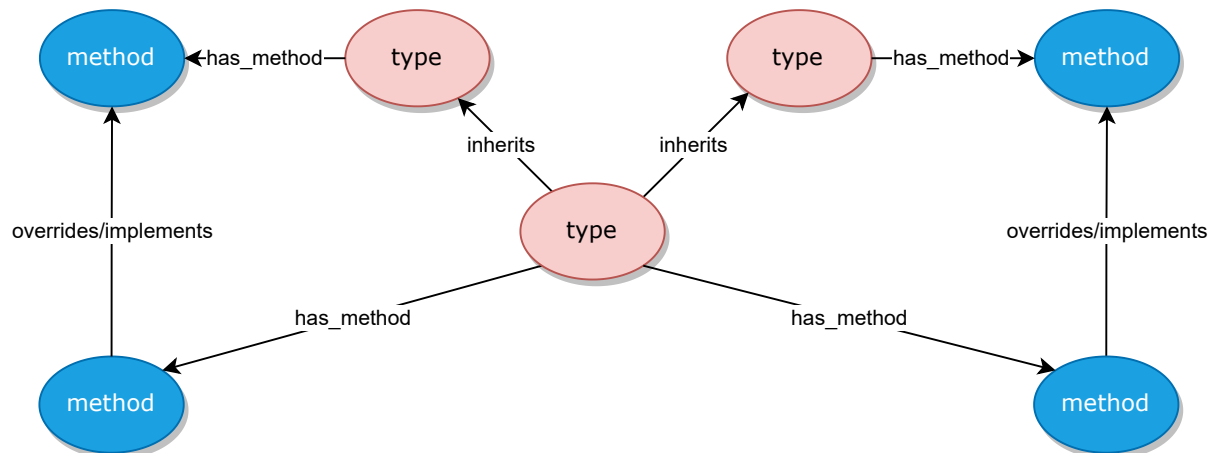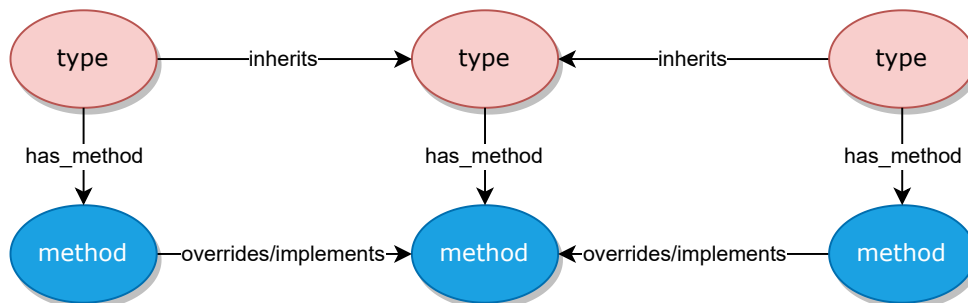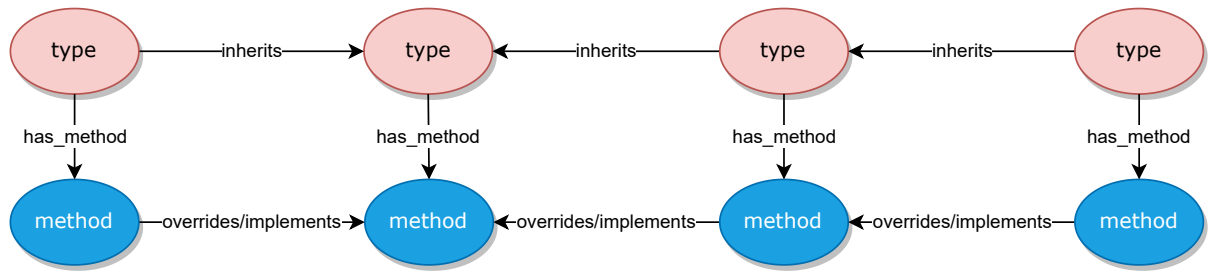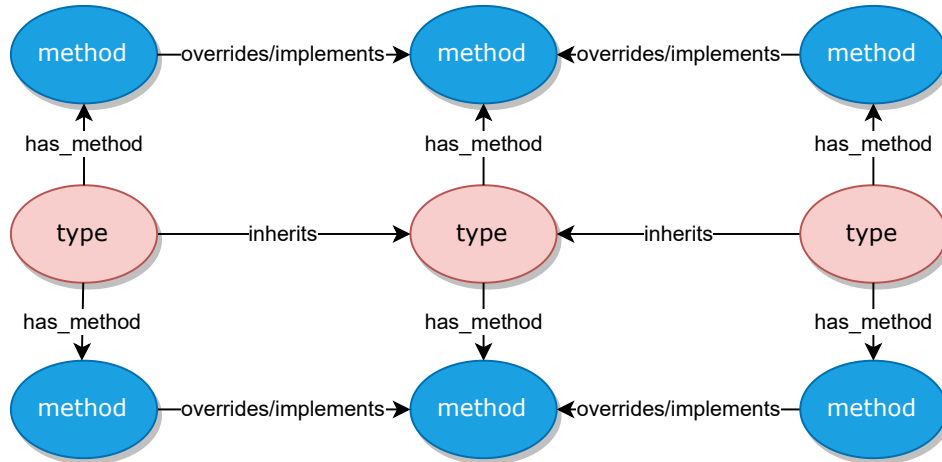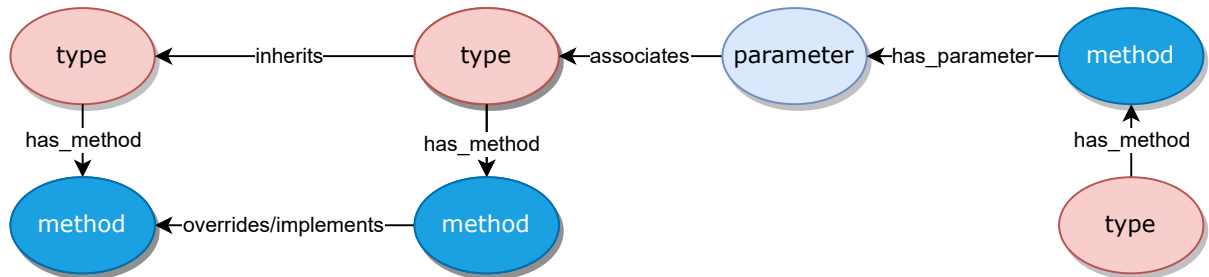
Composite, Proxy

$$(TMM \cup TM \cup TT)^3$$
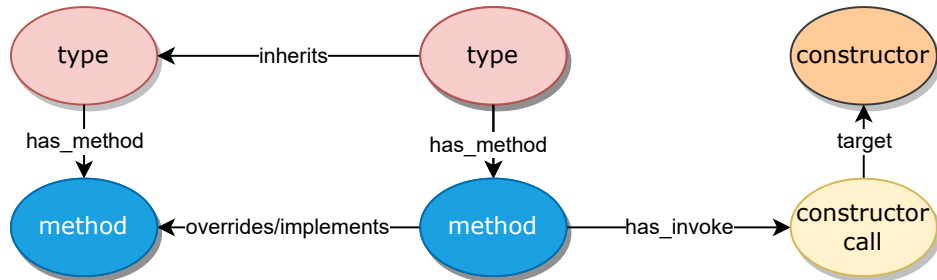
Decorator

$(TMM \cup TM \cup TT)^4$

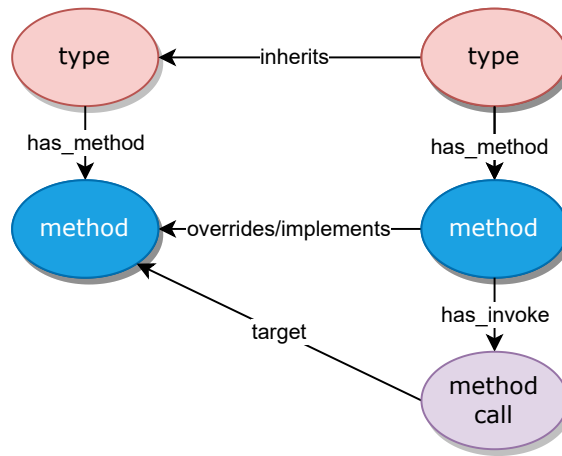Abstract Factory

$(TMM \cup TM \cup TT) \cup TMPT$

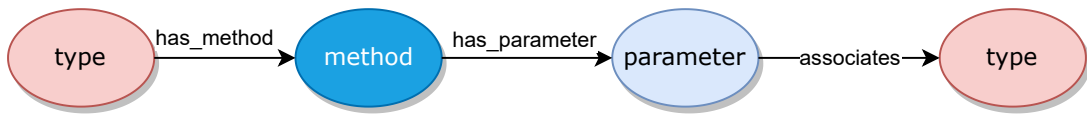*DPC − Factory parameter*

$(TMM \cup TM \cup TT) \cup TMI_cC$

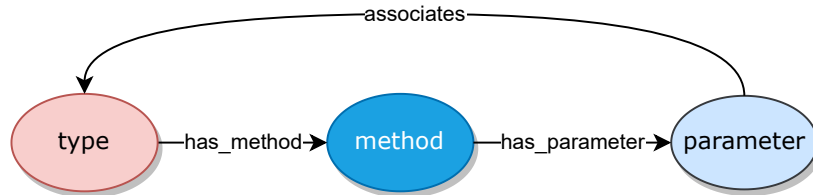*DPC − Factory Method*

$(TMM \cup TM \cup TT) \cup TMI_mM$

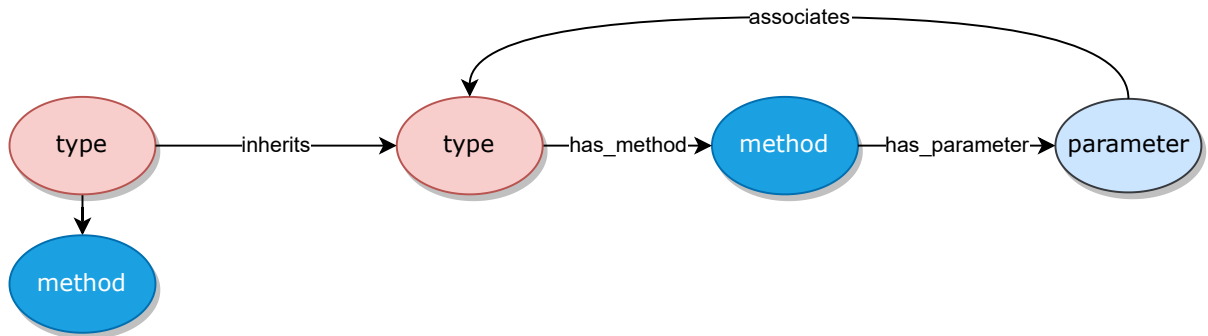*EDP − Revert Method, Extend Method*

TMPT

Memento, Visitor

DPC − Component method

$TMPT \cup TT \cup TM$

DPC − Leaf class, Node class

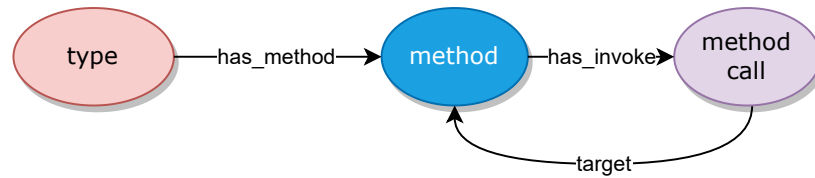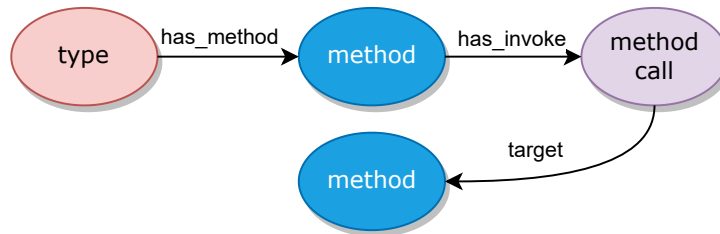$TMPT \cup TM \times 2 \cup TT$
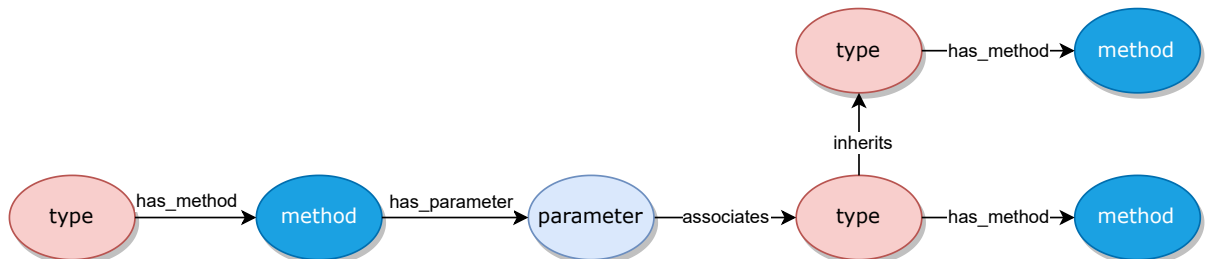
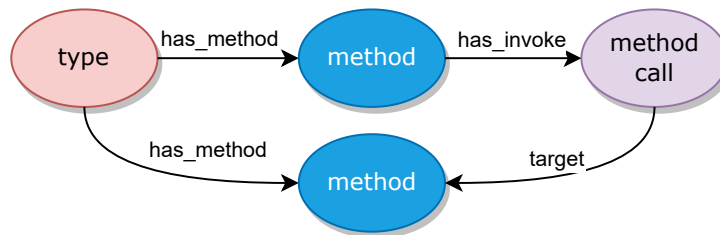TMI_mF
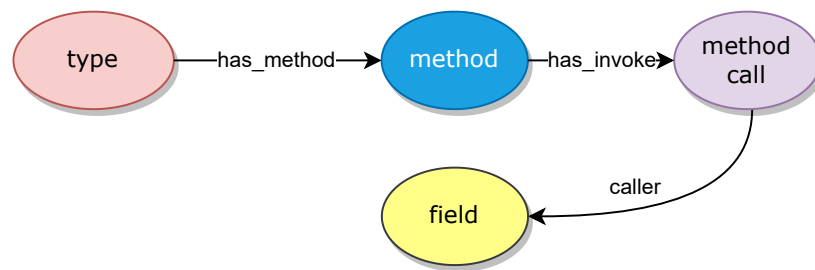
Observer

TMI_mM

EDP − Recursion

Chain of Responsibility

DPC − Abstract cyclic call
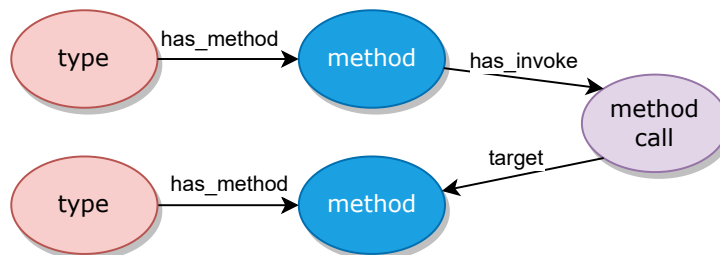
Adapter(Class), Decorator, Template Method

$TMI_mM \cup TM$

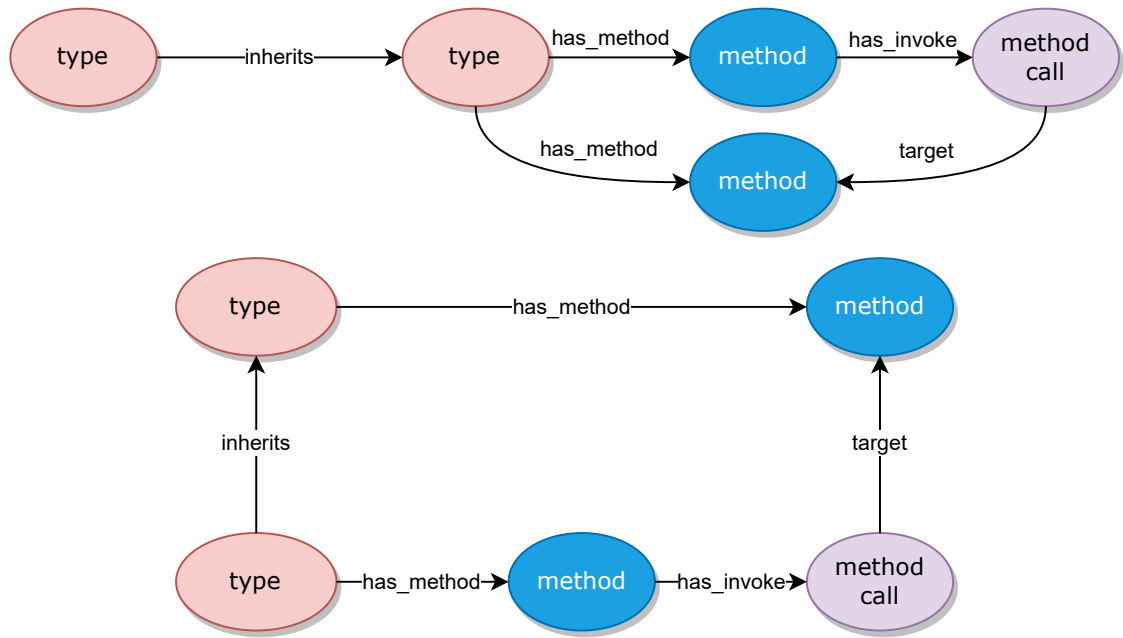*MP − Sink, Outline*
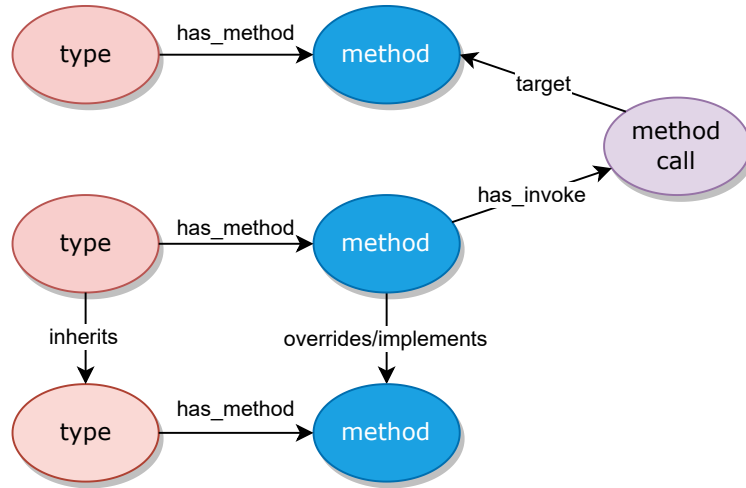*DPC − Template implementor, Template Method*
EDP − Conglomeration

DPC − Facade method, Multiple redirections in family, All methods invoked
EDP − Delegation, Redirection
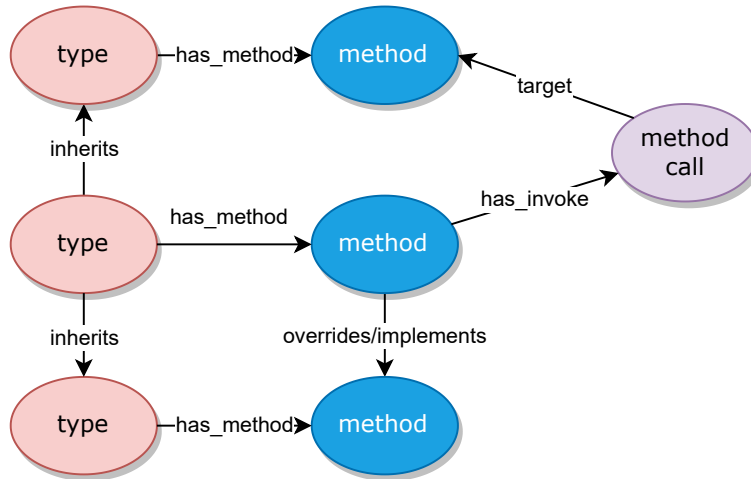
$TMI_mM \cup TT \cup TM$
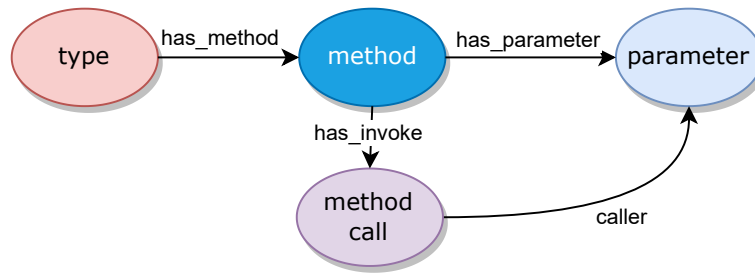
$$TMI_m M \cup TMM \cup TT \cup TM \times 2$$

DPC − Adapter method

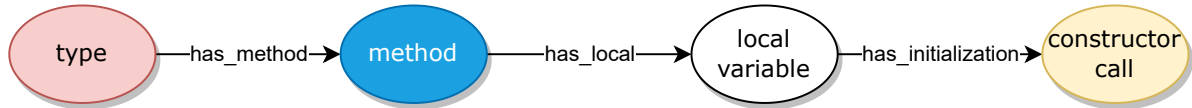$$TMI_m M \cup TMM \cup TT^2 \cup TM \times 2$$
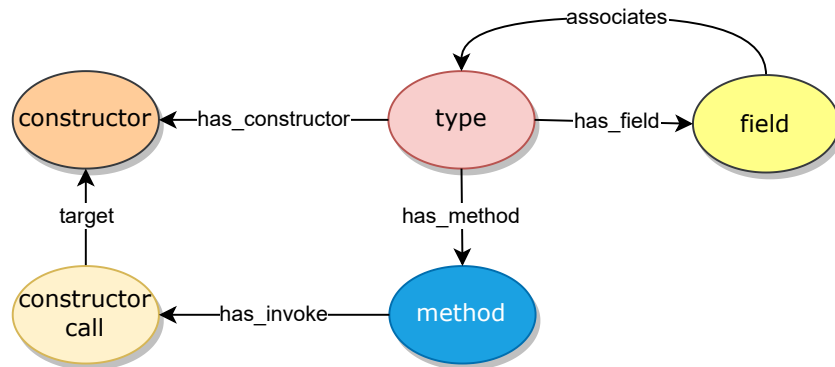

DPC − Adapter method
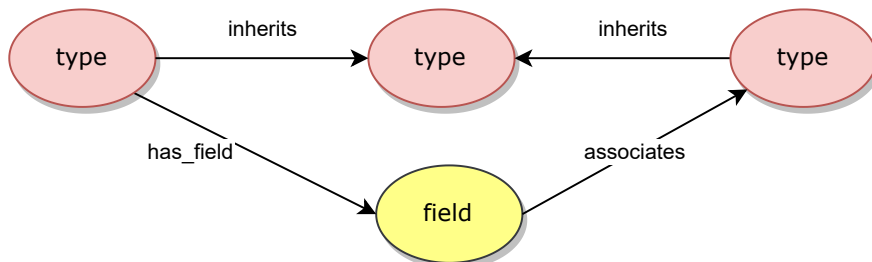
$$TMI_m P \cup TMP$$

$TMLI_c$

EDP − Create Object

$TMI_cC \cup TFT \cup TC$
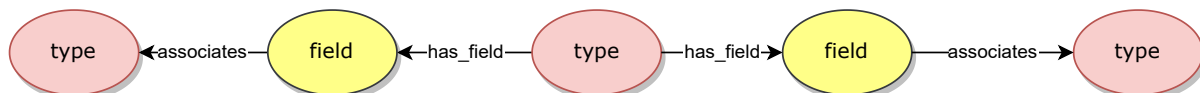

DPC − Controlled self instantiation

$TFT \cup TT^2$

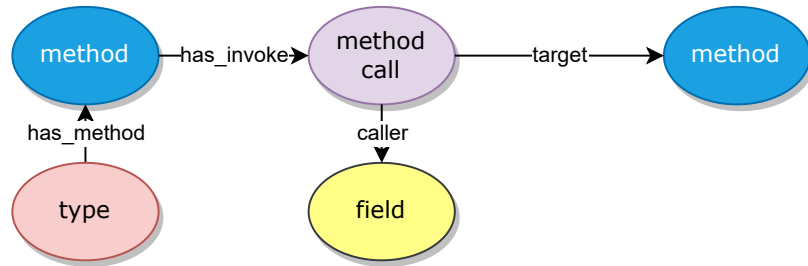*DPC − Proxy class, Same interface instance*

TFT^2

Flyweight

Mediator

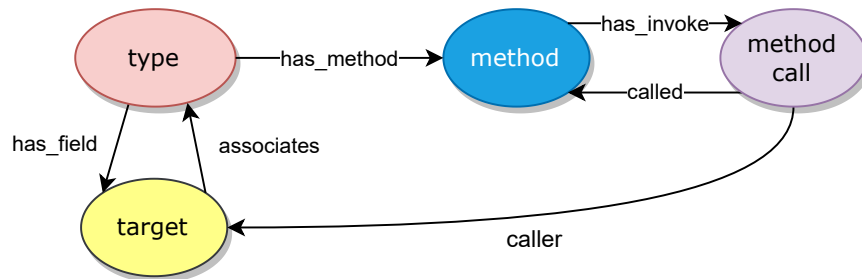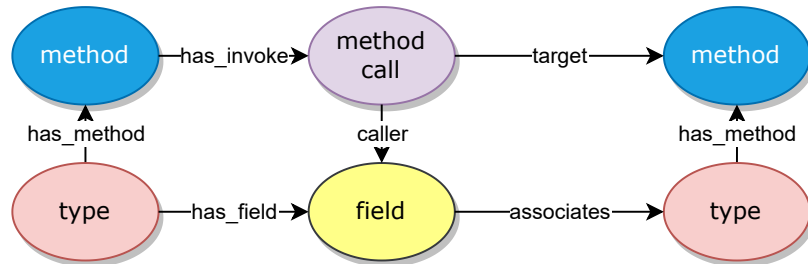# 5-order

$TMI_mF \cup TMI_mM$

Builder, Prototype, Adapter(object), Bridge, Composite, Decorator, Proxy, Command, Observer, State, Strategy
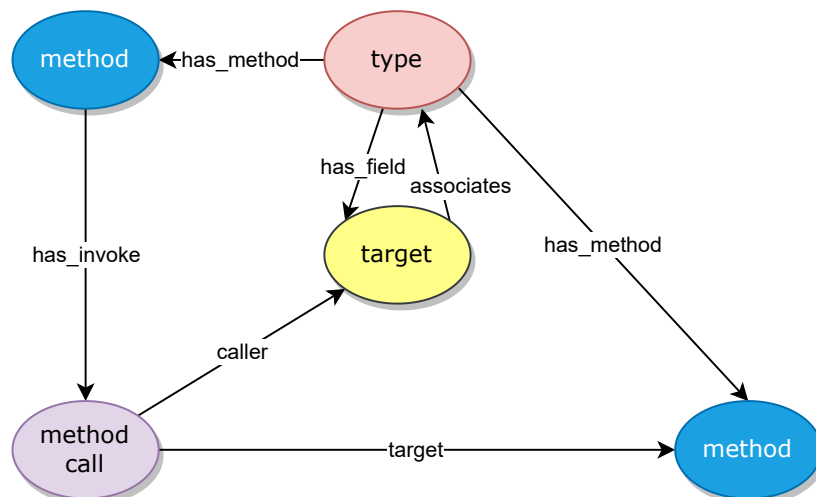
$(TMI_mF \cup TMI_mM) \cup TFT$

$EDP - Redirected\ Recursion$
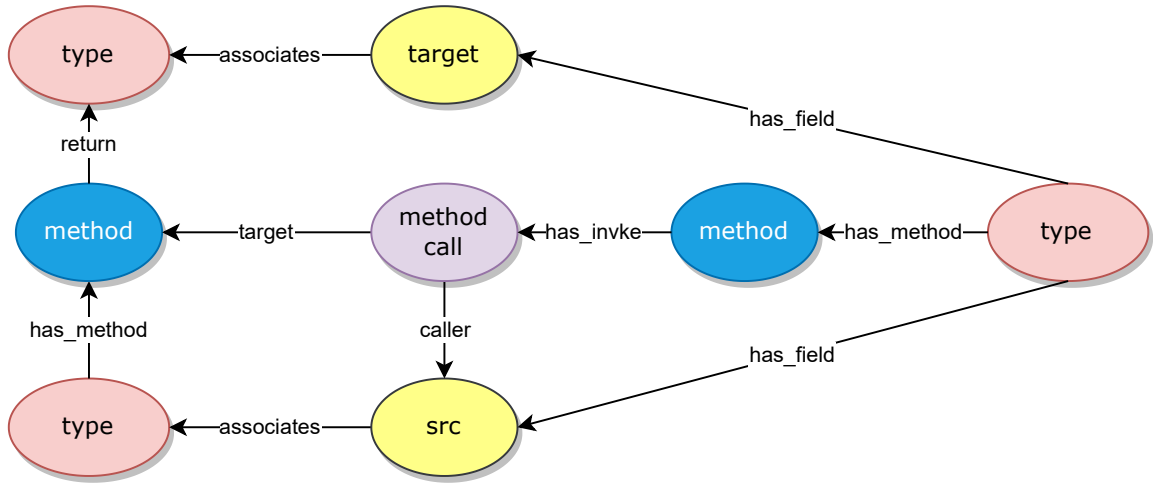
$(TMI_mF \cup TMI_mM) \cup TFT \cup TM$

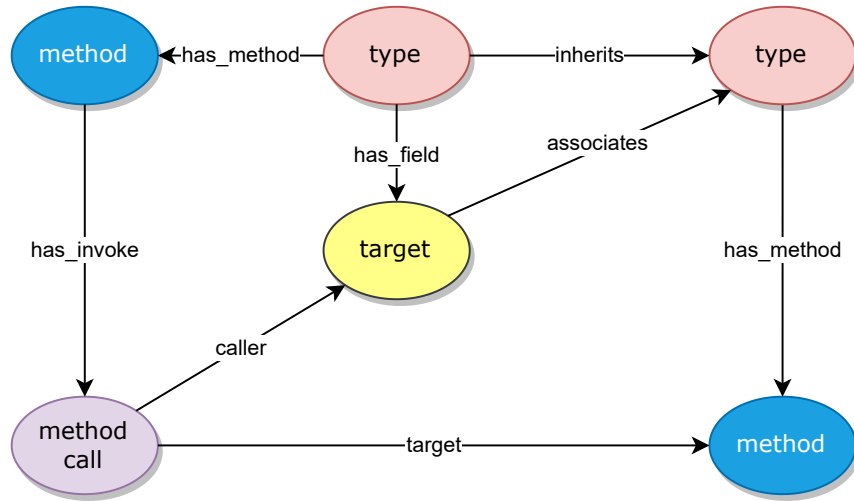$DPC - Instance\ in\ abstract\ referred$

$EDP - Delegated\ Conglomeration$

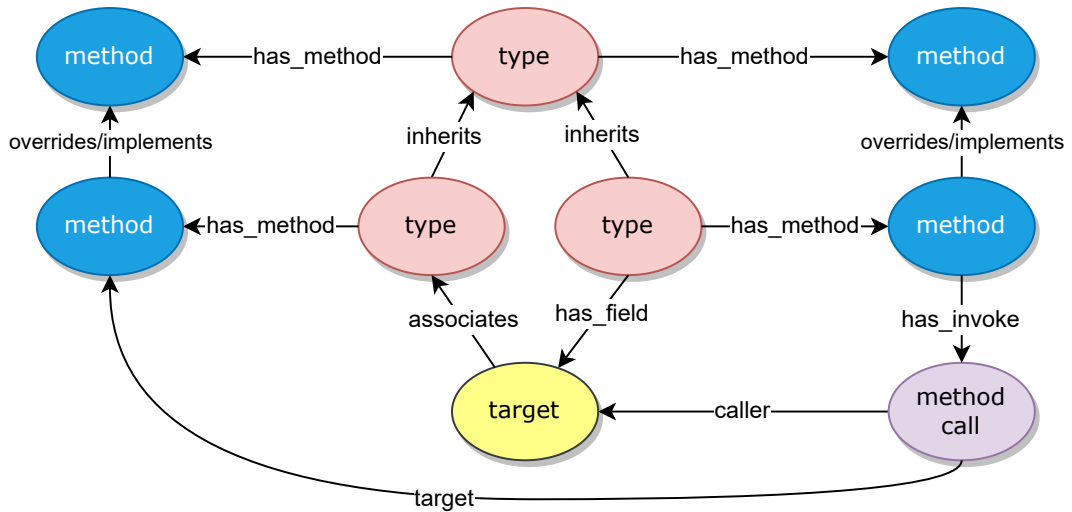$(TMI_mF \cup TMI_mM) \cup TFT^2 \cup TMT$

EDP − Retrieve

$(TMI_mF \cup TMI_mM) \cup TFT \cup TM \cup TT$
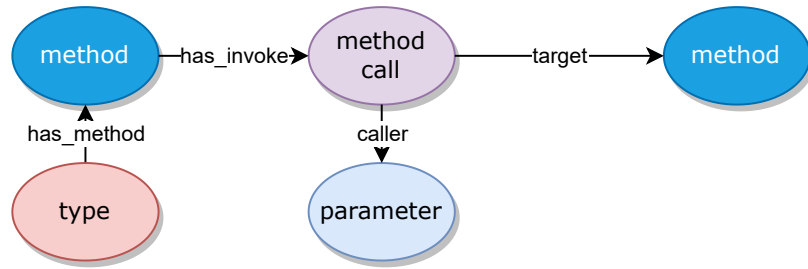

EDP − Trusted Delegation, Trusted Redirection

$(TMI_mF \cup TMI_mM) \cup (TMM \cup TT \cup TM)^2 \cup TFT$
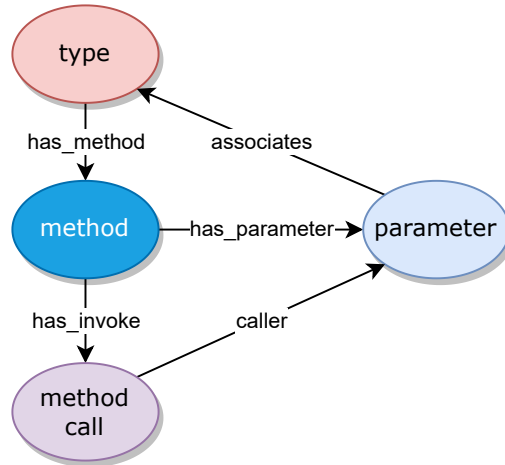

DPC − Proxy method invoked
EDP − Deputized Delegation, Deputized Redirection
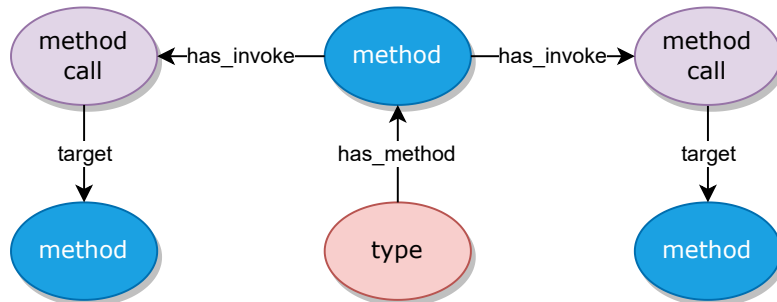
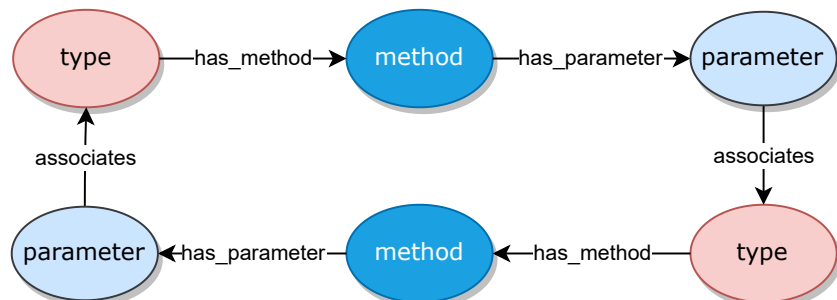$TMI_mP \cup TMI_mM$

Memento, Visitor
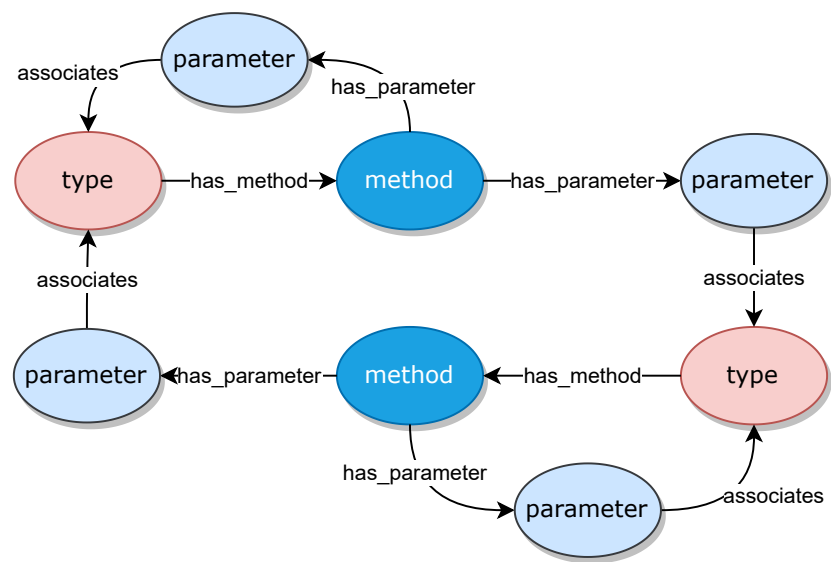
$TMI_mP \cup TMPT$

$DPC - This\ parameter$

TMI_mM^2

Template Method

$TMPT^2$

$DPC - Cross\ relationship$

$TMPT^4$

DPC − Visitable class

6-order

$(TMI_mF \cup TMI_mM)^2$

Builder