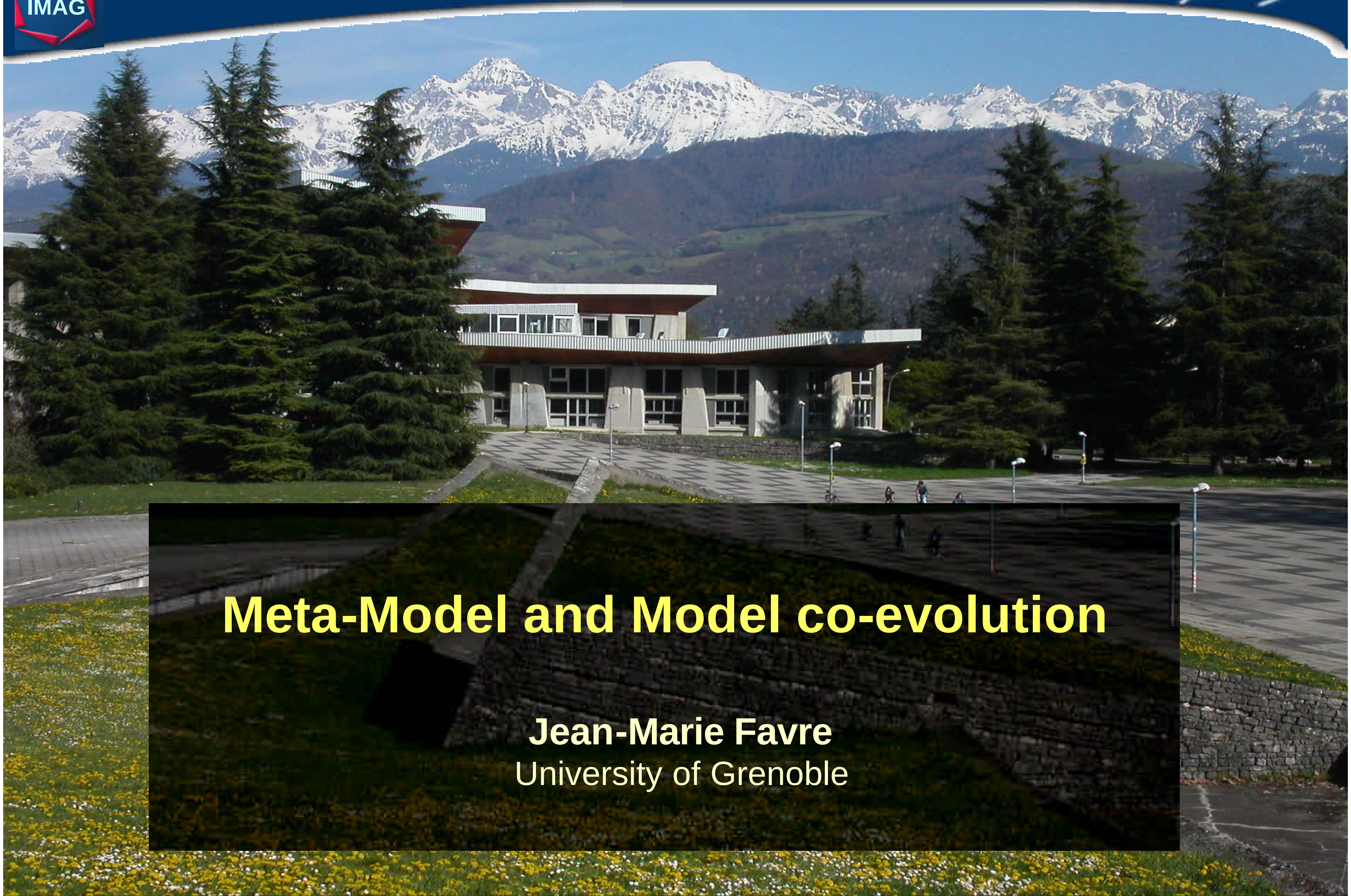
IMAG

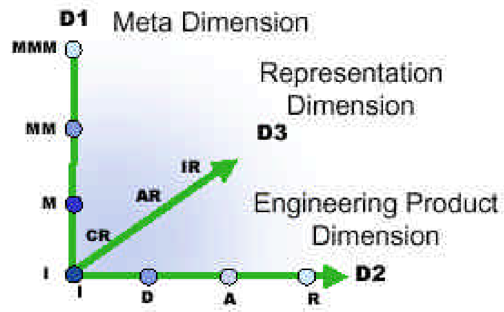


Meta-Model and Model co-evolution

Jean-Marie Favre
University of Grenoble

OUTLINE

- **Motivation and background : Industry**
- **Software in 3D**
  - D1: meta dimension
  - D2: engineering dimension
  - D3: representation dimension



- **Evolution: entering the 4th dimension…**
- **Conclusion**

Part I :

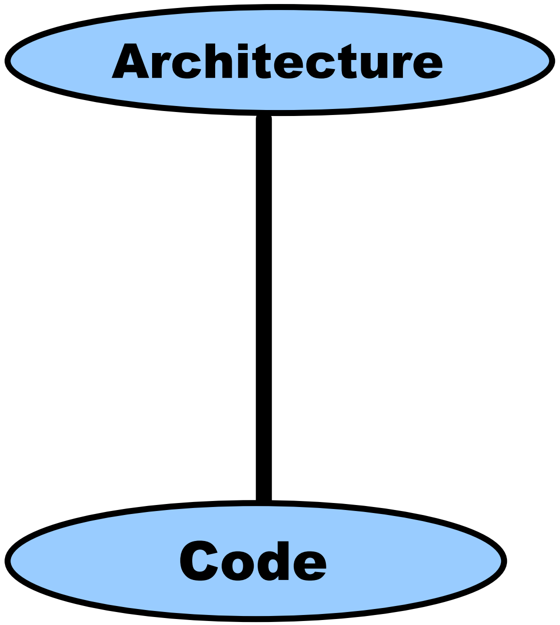
Motivation and Background

Historical mistakes in Software Engineering

- (1) Software is stable
- (2) Software is made of programs

Everything evolve in complex industrial contexts

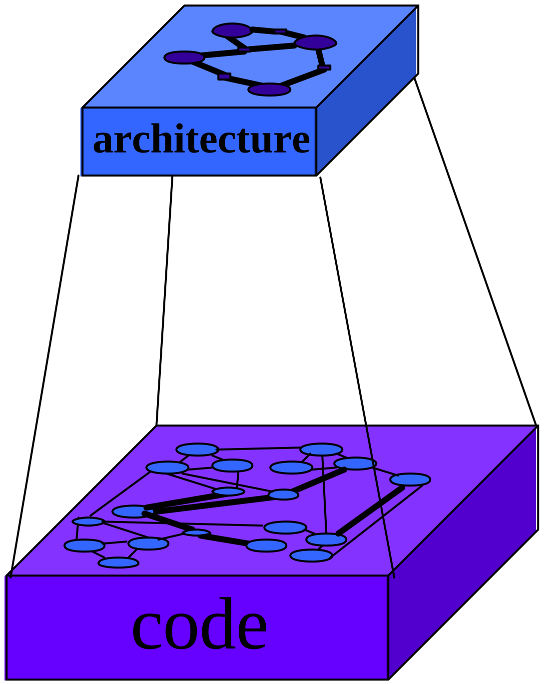
Architecture and Code co-evolution



- Explicit vs. implicit architecture
- Architecture and code both evolve
- Horizontal impacts
- Vertical impacts
- Synchronization and conformance issues
- Risks of erosion
- Architecture-driven vs. code-driven

- A "well identified" phenomenon nowadays
- Initially neglected by academics

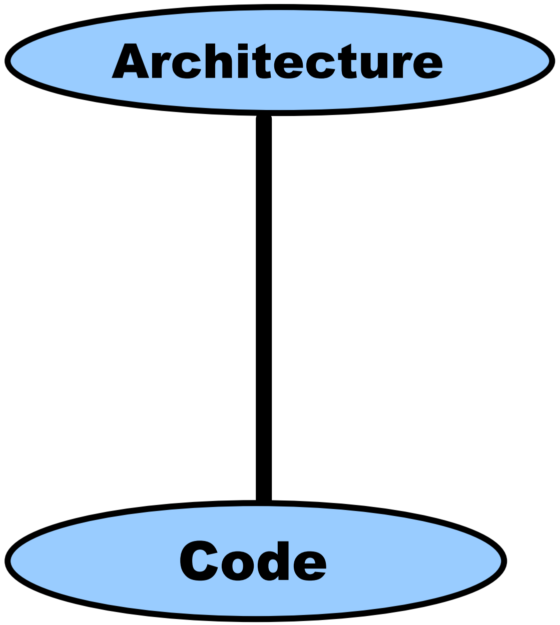
Architecture and Code co-evolution



- Explicit vs. implicit architecture
- Architecture and code both evolve
- Horizontal impacts
- Vertical impacts
- Architecture-driven vs. code-driven
- Synchronization and conformance issues
- Risks of erosion

- A "well identified" phenomenon nowadays
- Initially neglected by academics

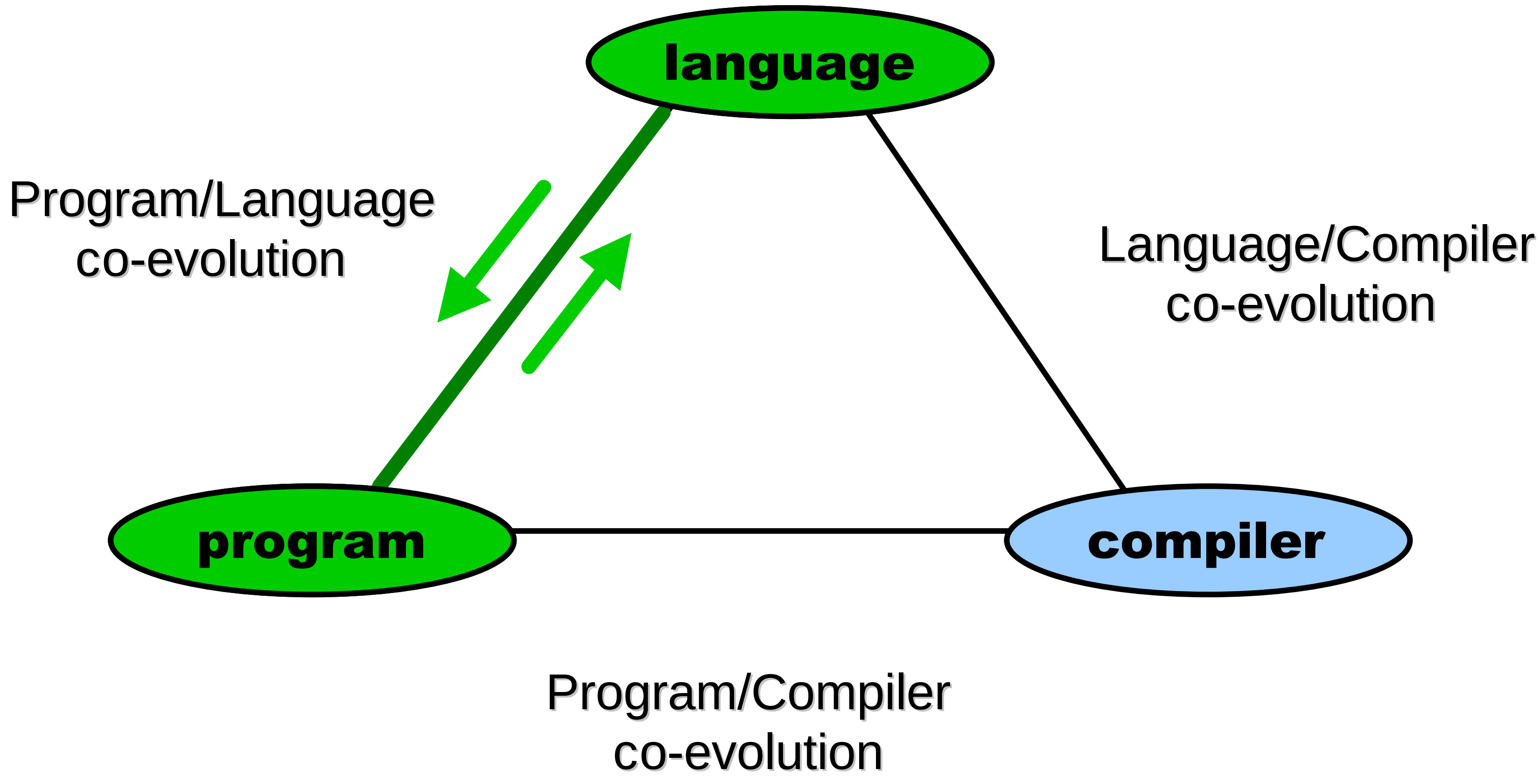
Architecture and Code co-evolution



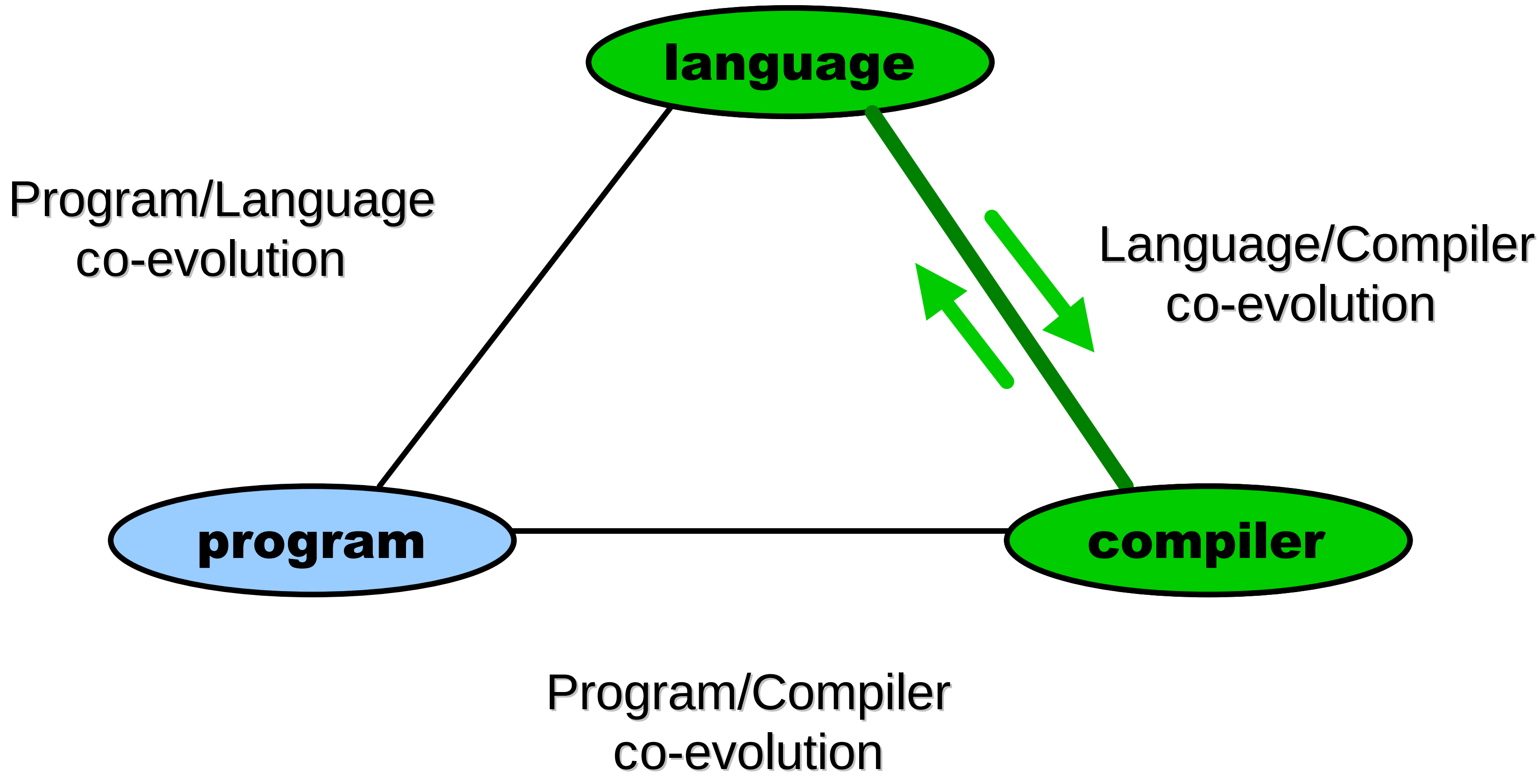
- Explicit vs. implicit architecture
- Architecture and code both evolve
- Horizontal impacts
- Vertical impacts
- Synchronization and conformance issues
- Risks of erosion
- Architecture-driven vs. code-driven

- A "well identified" phenomenon nowadays
- Initially neglected by academics

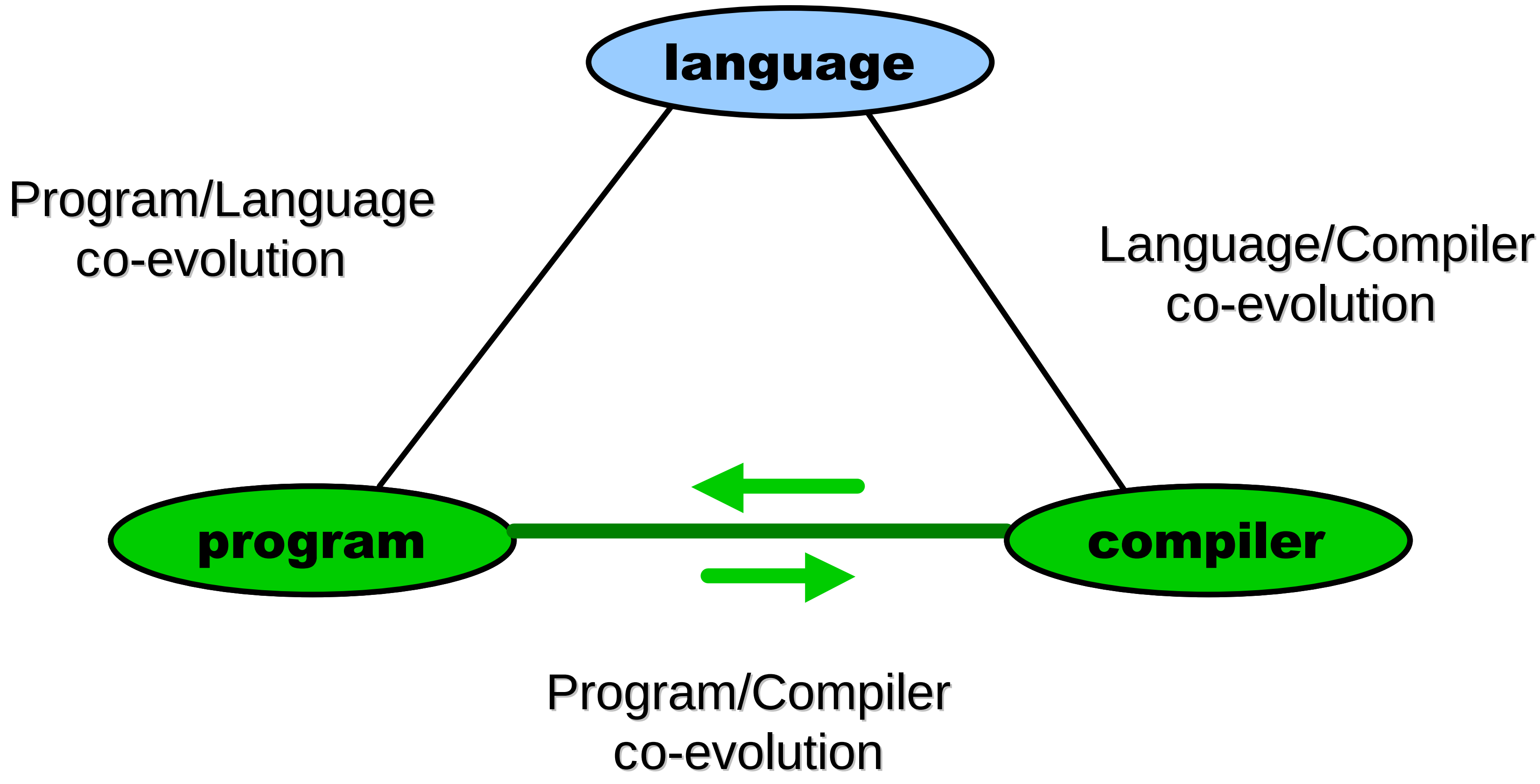
Program / Language / Tool co-evolution



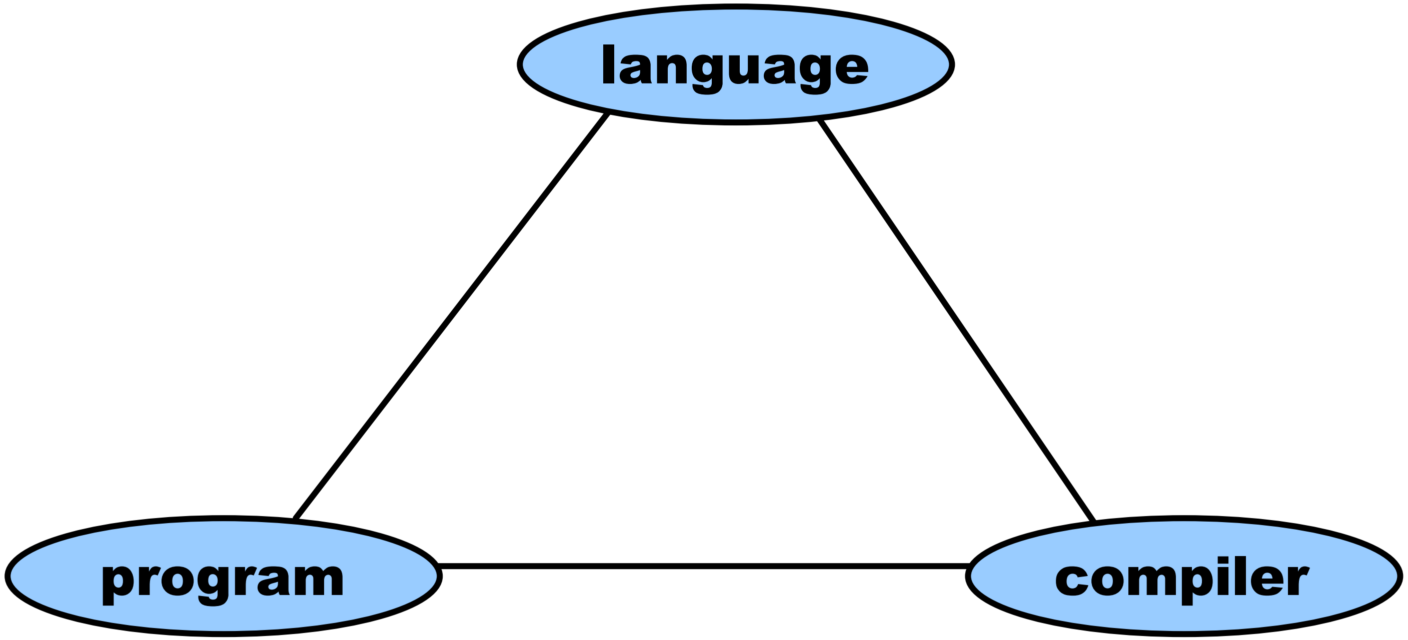
Program / Language / Tool co-evolution



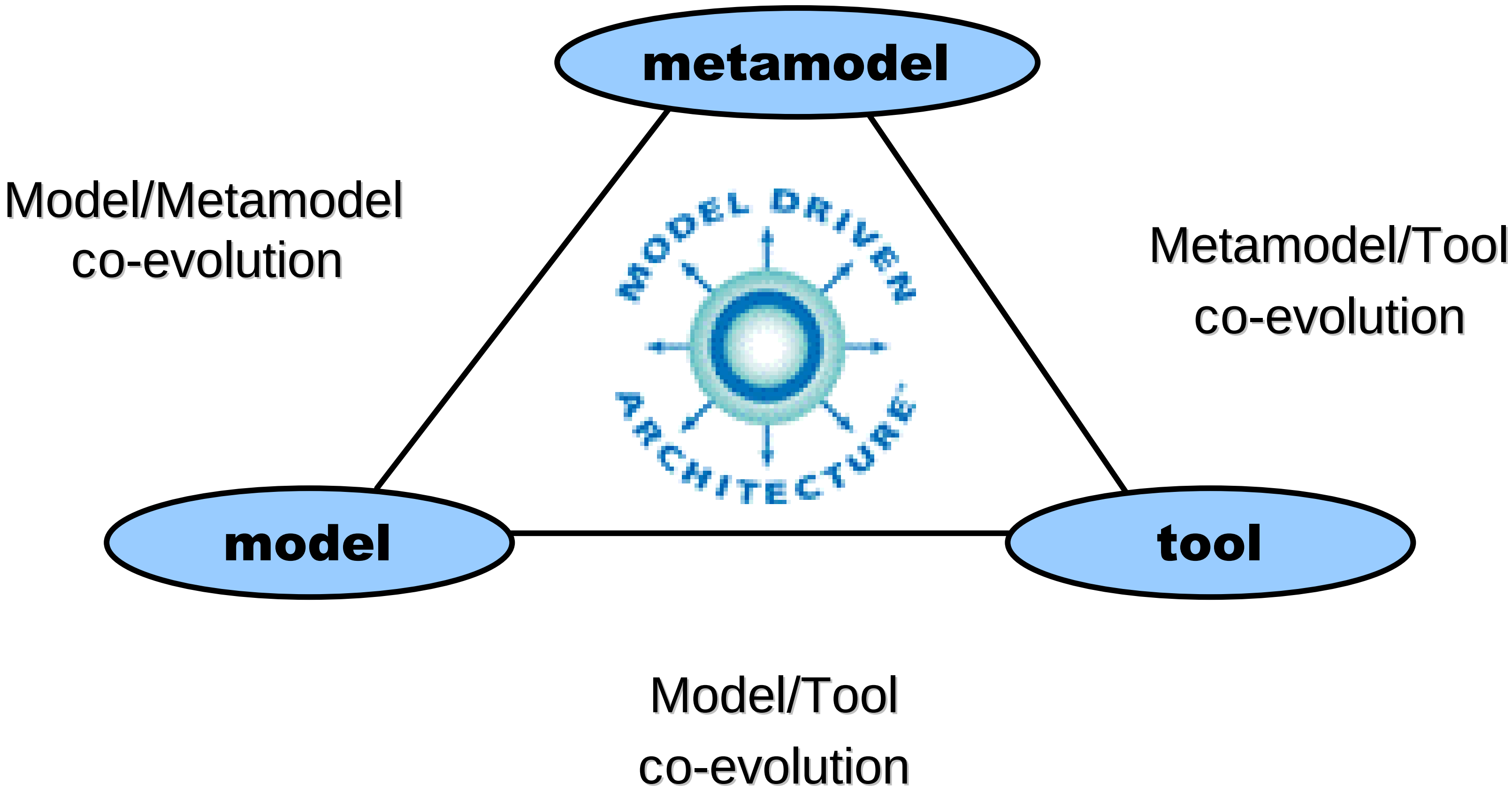
Program / Language / Tool co-evolution



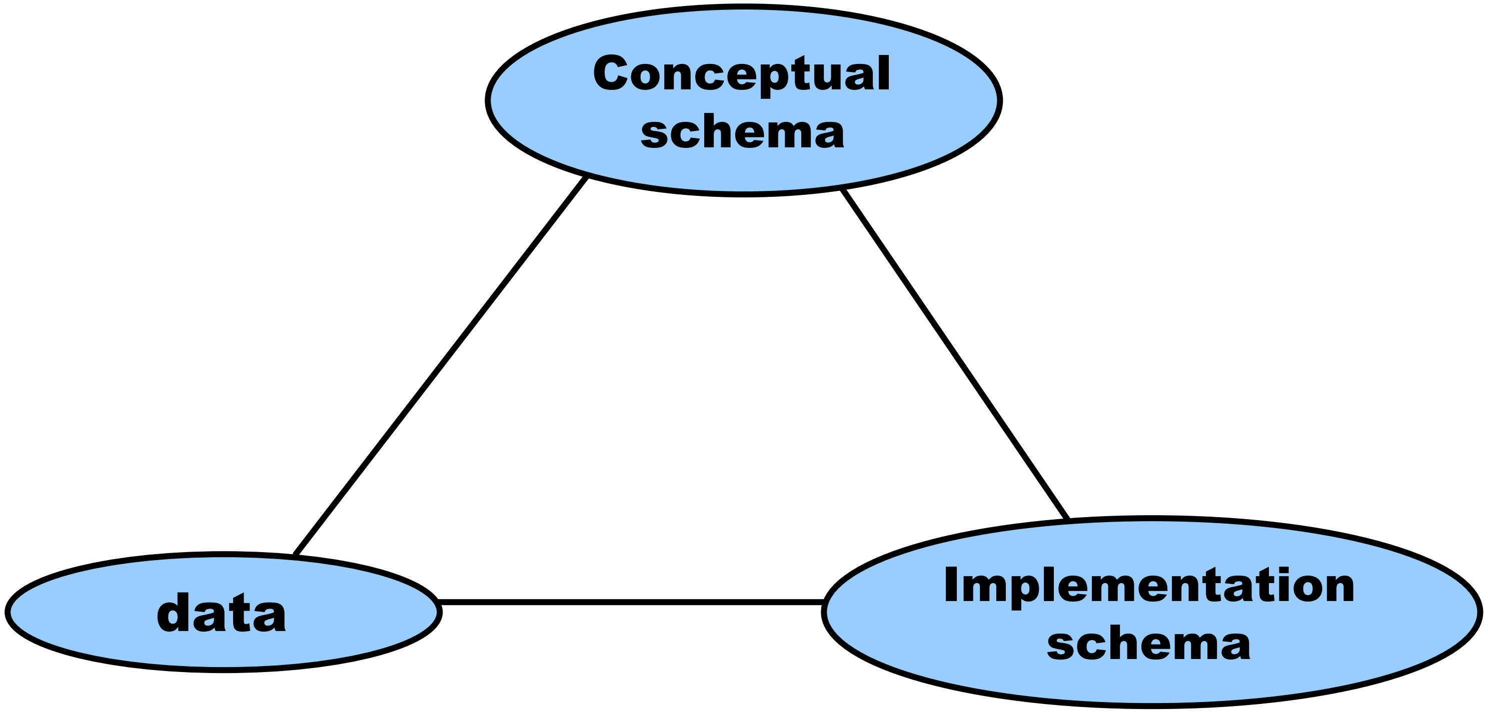
Program / Language / Tool co-evolution

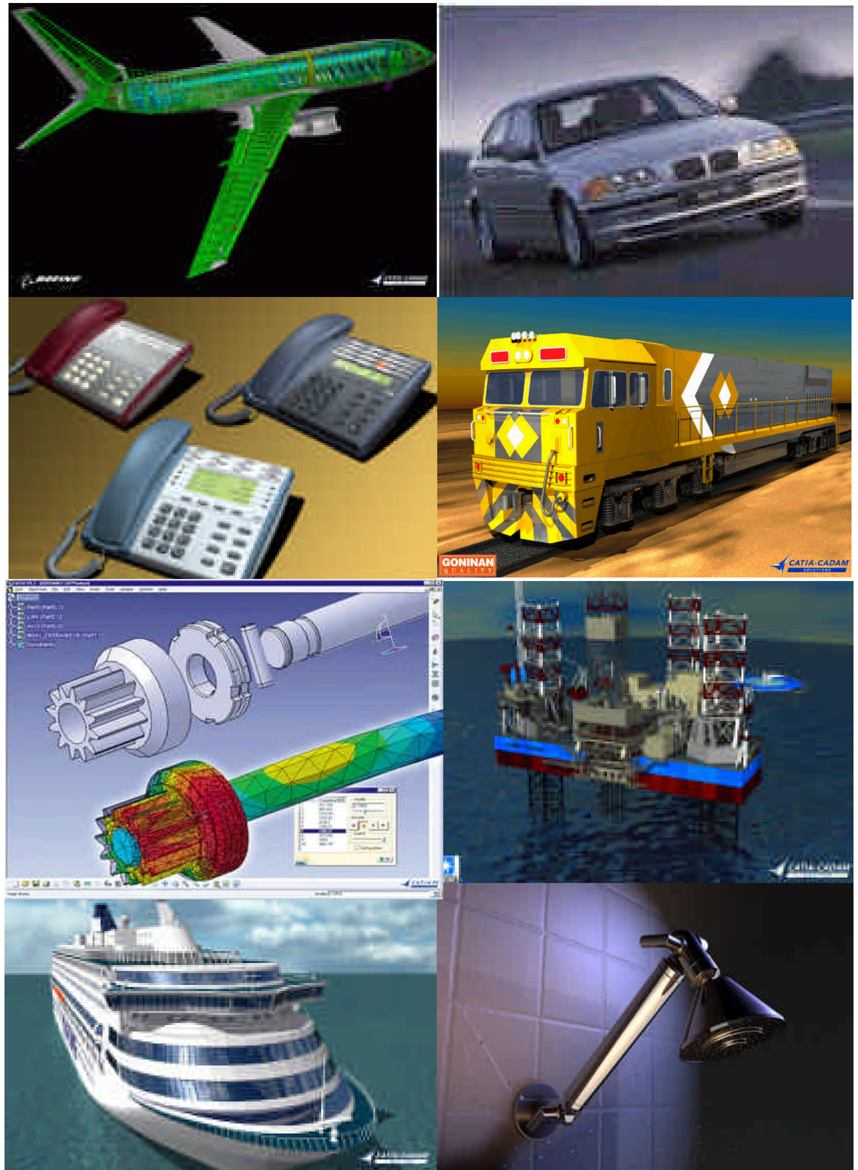


Model / Meta-Model / Tool co-evolution



Schema Evolution



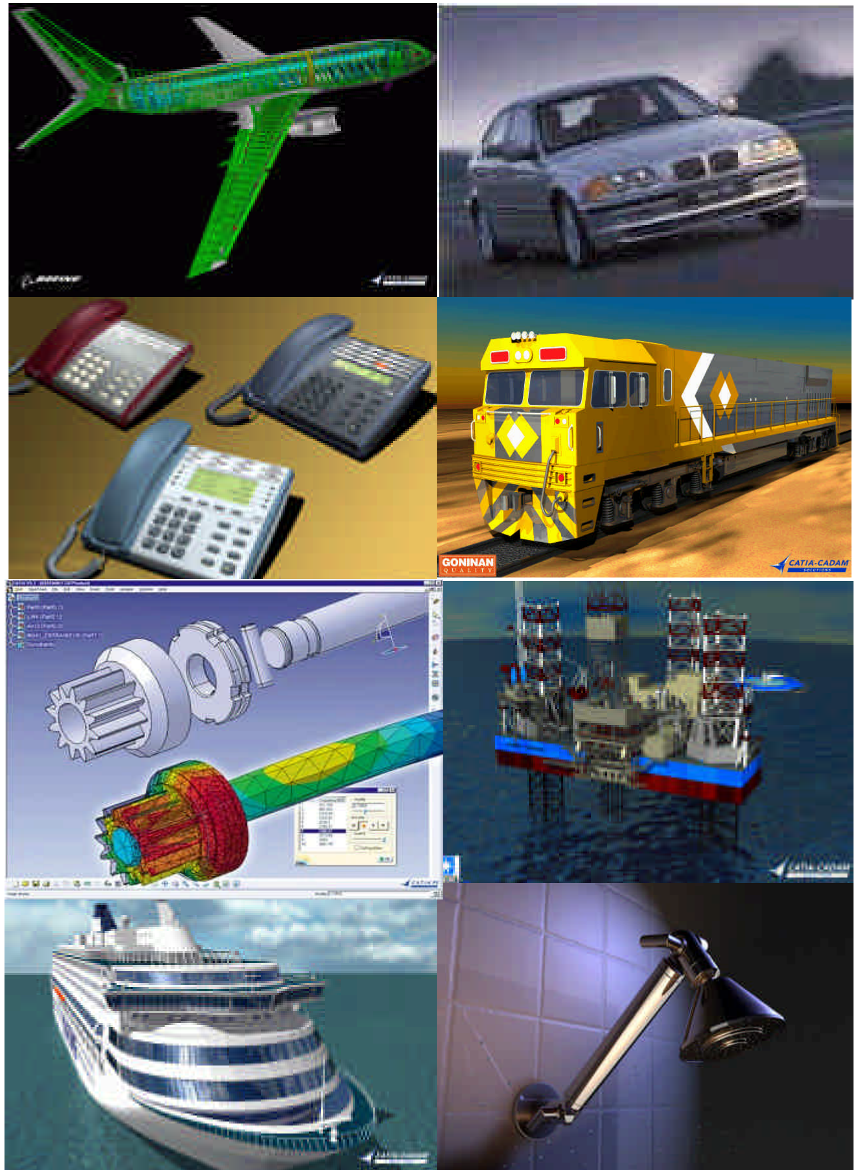


Background : A 7-year case study

Collaboration with industry

DASSAULT SYSTEMES

- World leader in CAD/CAM
- 19 000 clients, 180 000 seats
- Clients: Boeing, Chrysler, …
- Main software: CATIA



CATIA: a very large Software Product Line

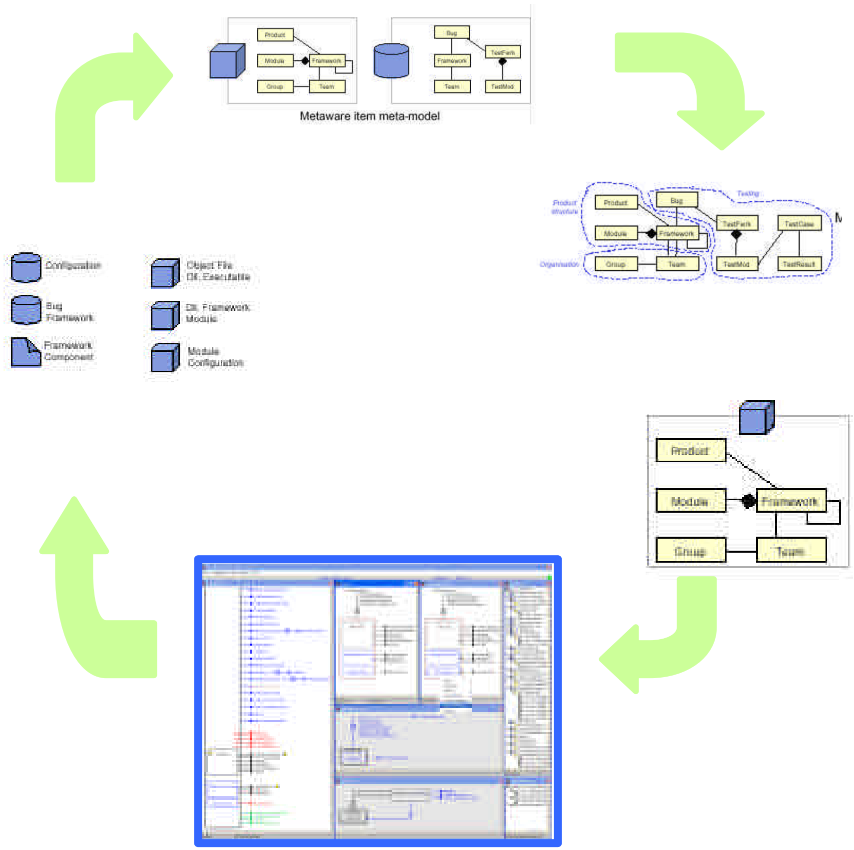
- 1200+ software engineers
- 70 000+ classes C++
- 8 000+ components
- 5 000+ interfaces
- 3 000+ DLLs
- 800+ frameworks
- …

Need to raise the level<u>s</u> of abstraction

- **Architecture**
- **Metamodel**

A Meta-Model Driven Architecture Recovery Process

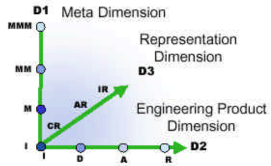
Metaware domain and asset analysis	Metaware inventory Meta-models recovery Meta-models Integration Meta-model clustering Meta-model packaging
Metaware requirement analysis	Meta-level actors identification Meta-level use cases identification Metaware assesment Metaware Improvement analysis Meta-level use cases description
Metaware specification	Meta-model filtering and extension Presentation specification Metaware specification packaging
Metaware implementation	Extractors development and reuse Viewers development and reuse Extractors and Views integration
Metaware execution	Execution
Metaware evolution	Evaluation Feedback

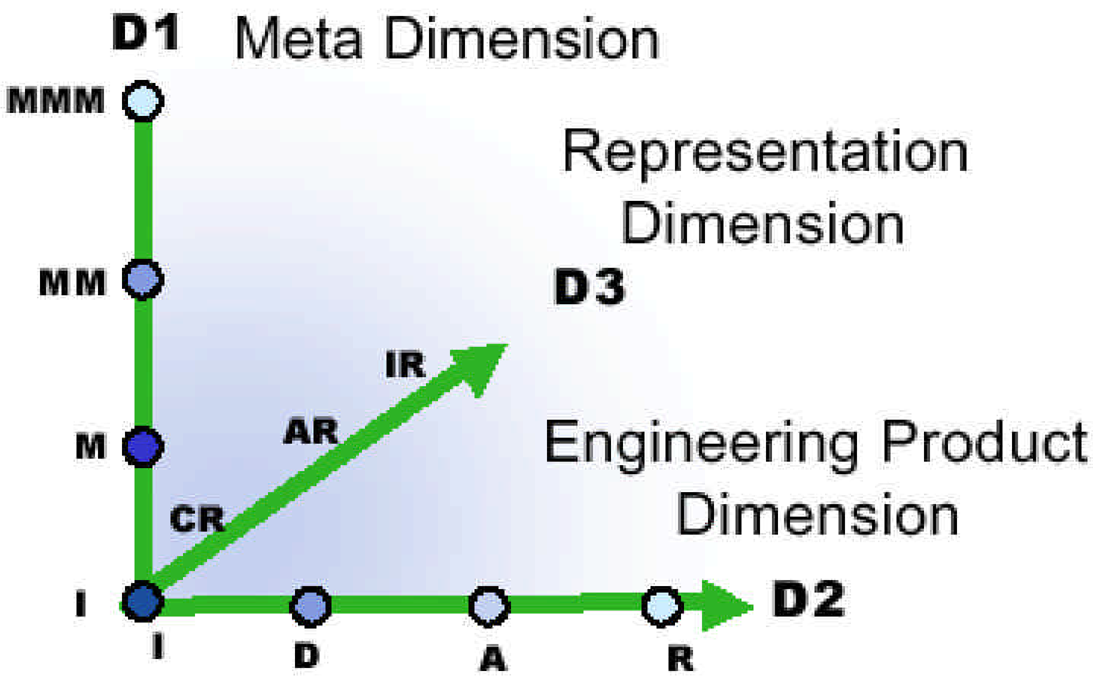


Part II :

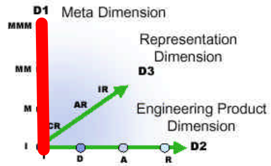
The 3D Software Space

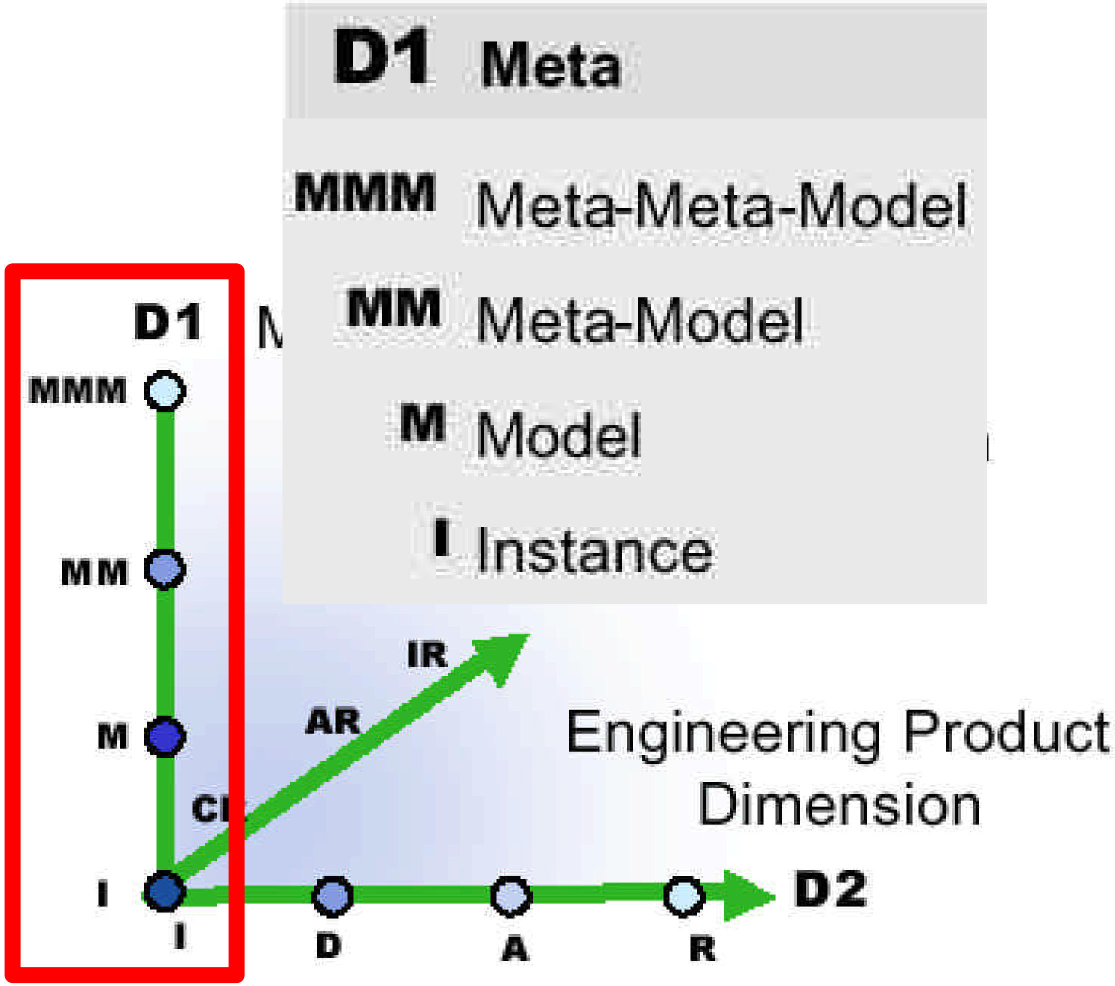
The 3D Software space



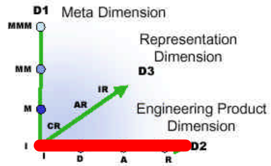


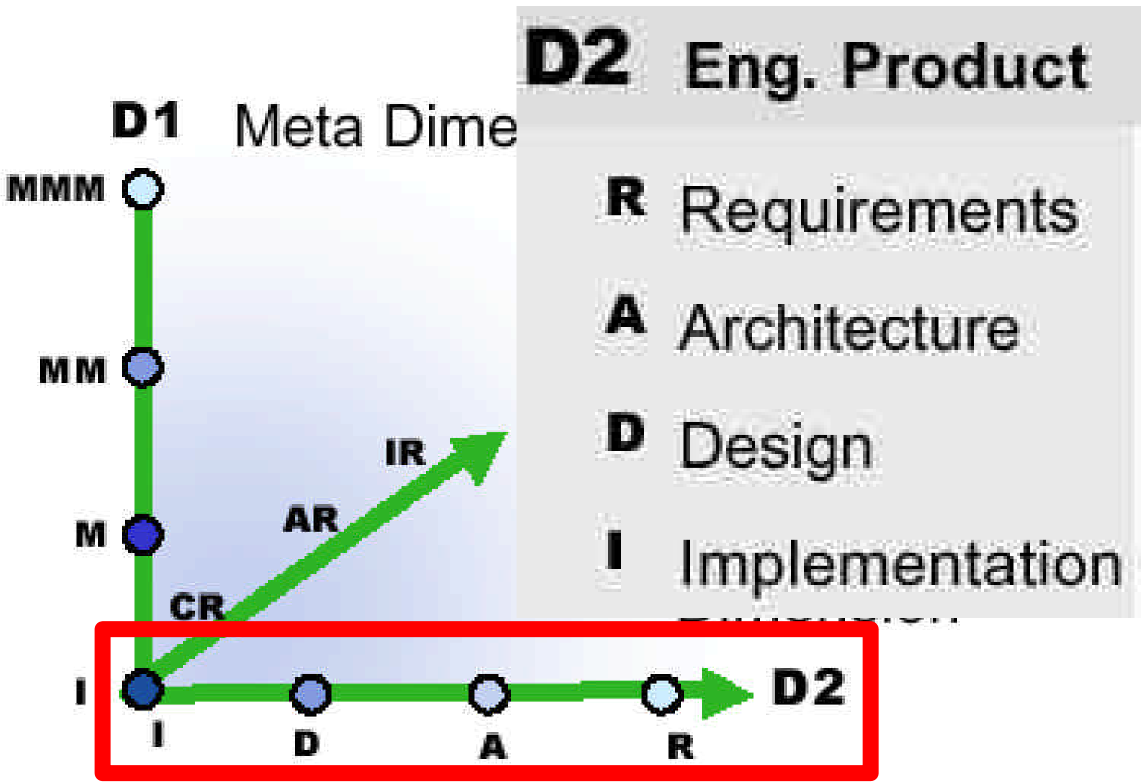
The 3D Software space



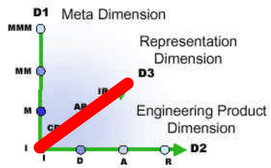


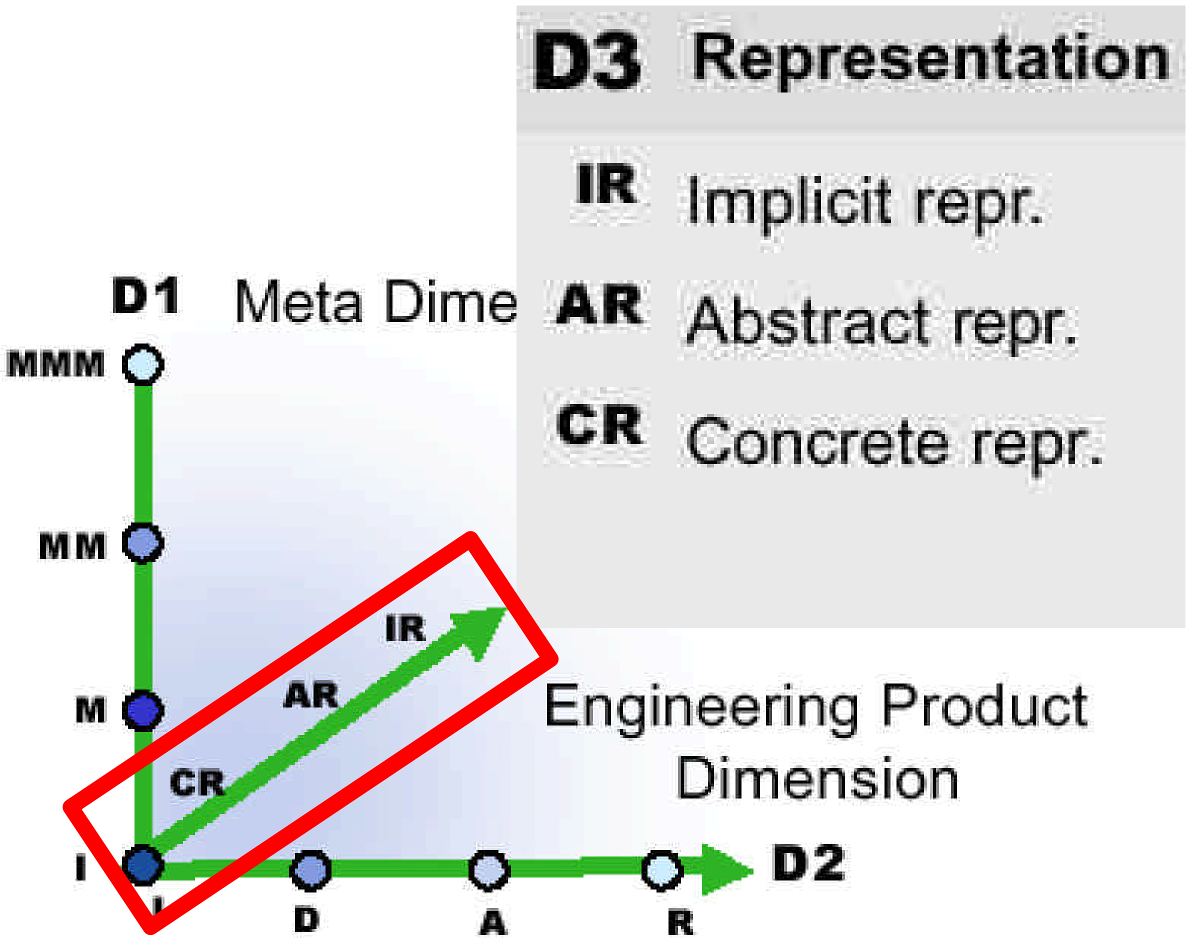
The 3D Software space



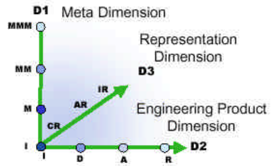


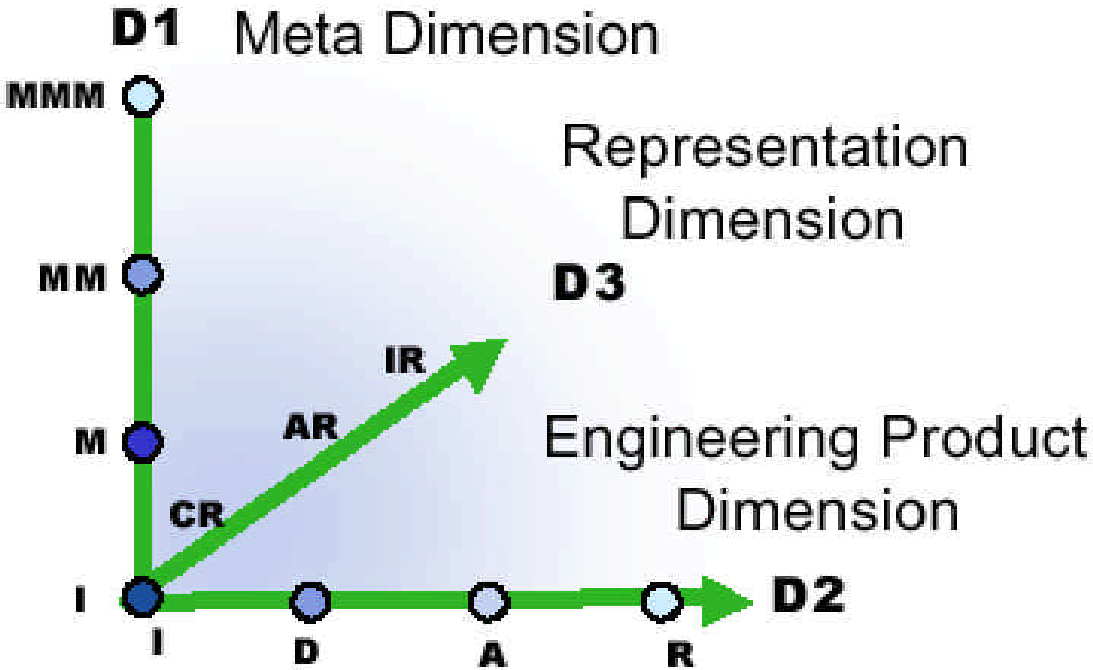
The 3D Software space



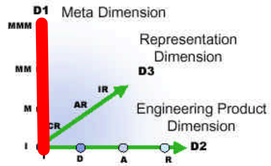


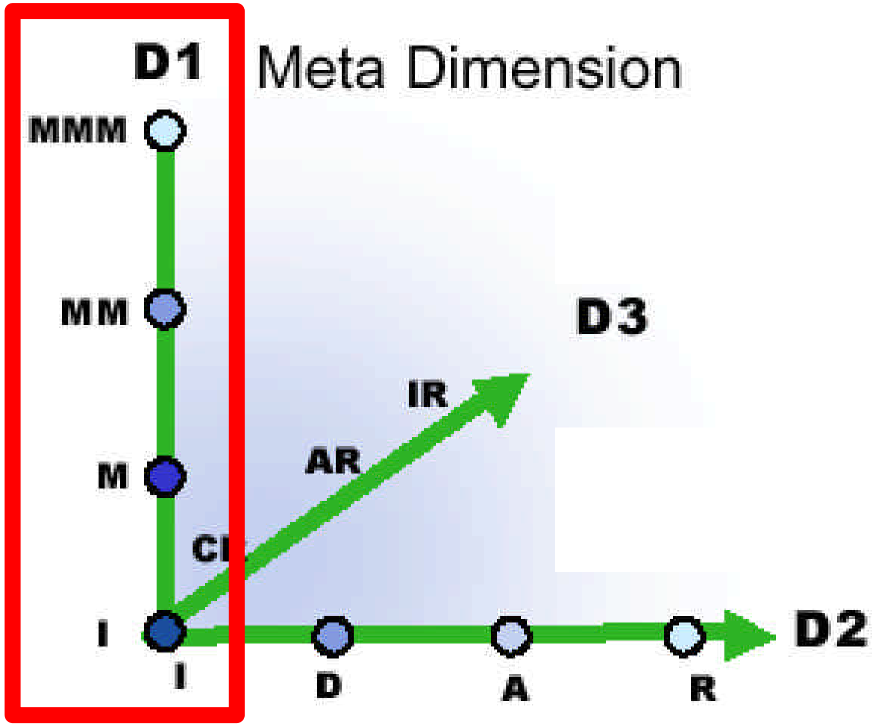
A taxonomy of software artefacts





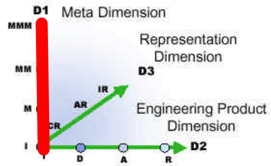
D1: The Meta dimension

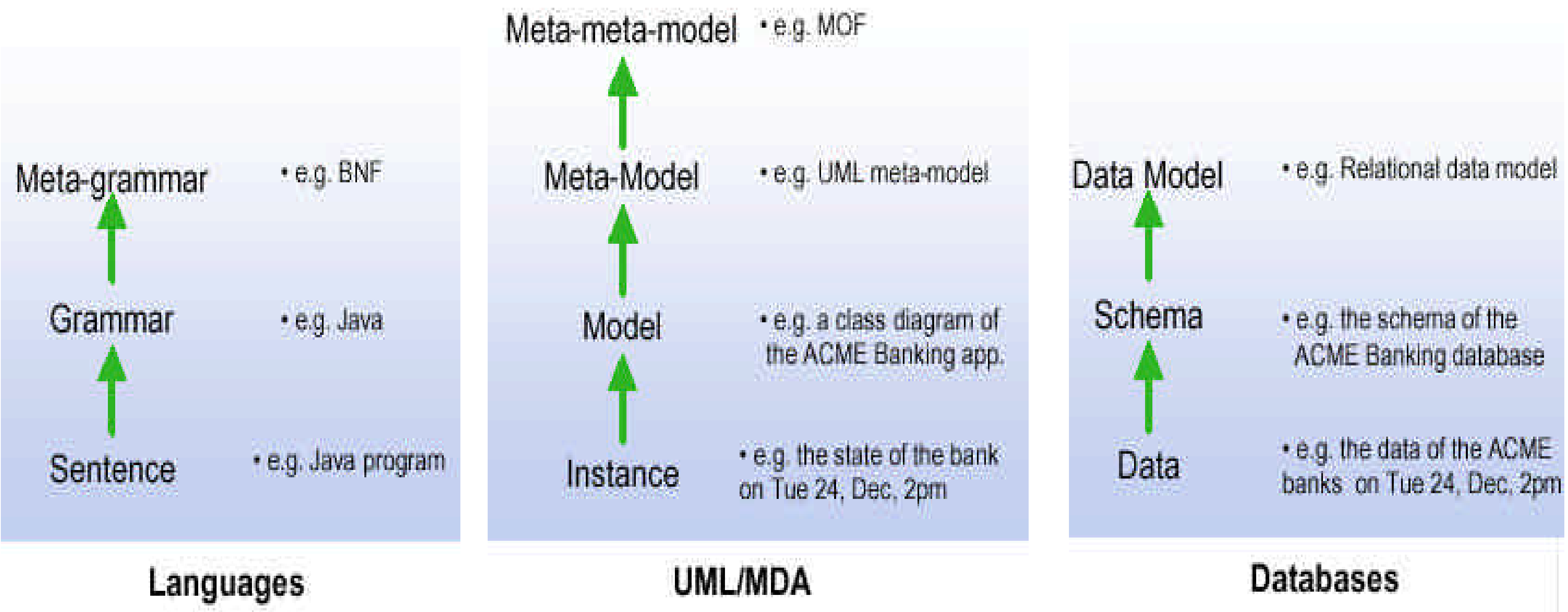




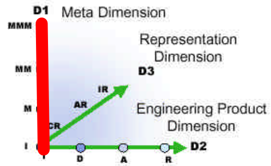
- The Meta-towers
- The Meta-pyramid
- The Meta actor pyramid

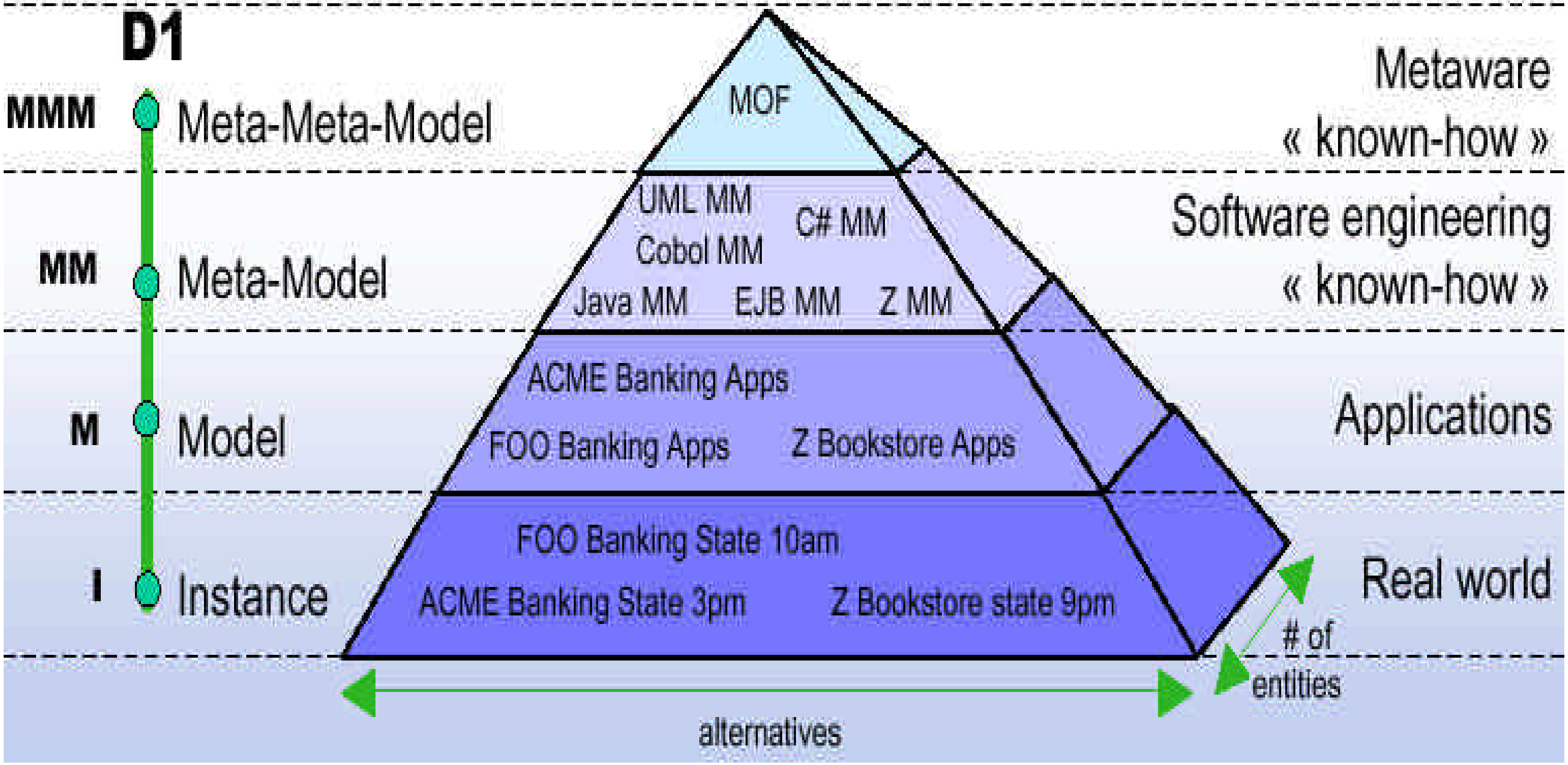
D1: The Meta-towers



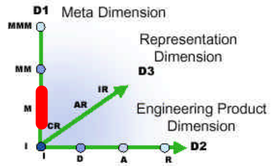


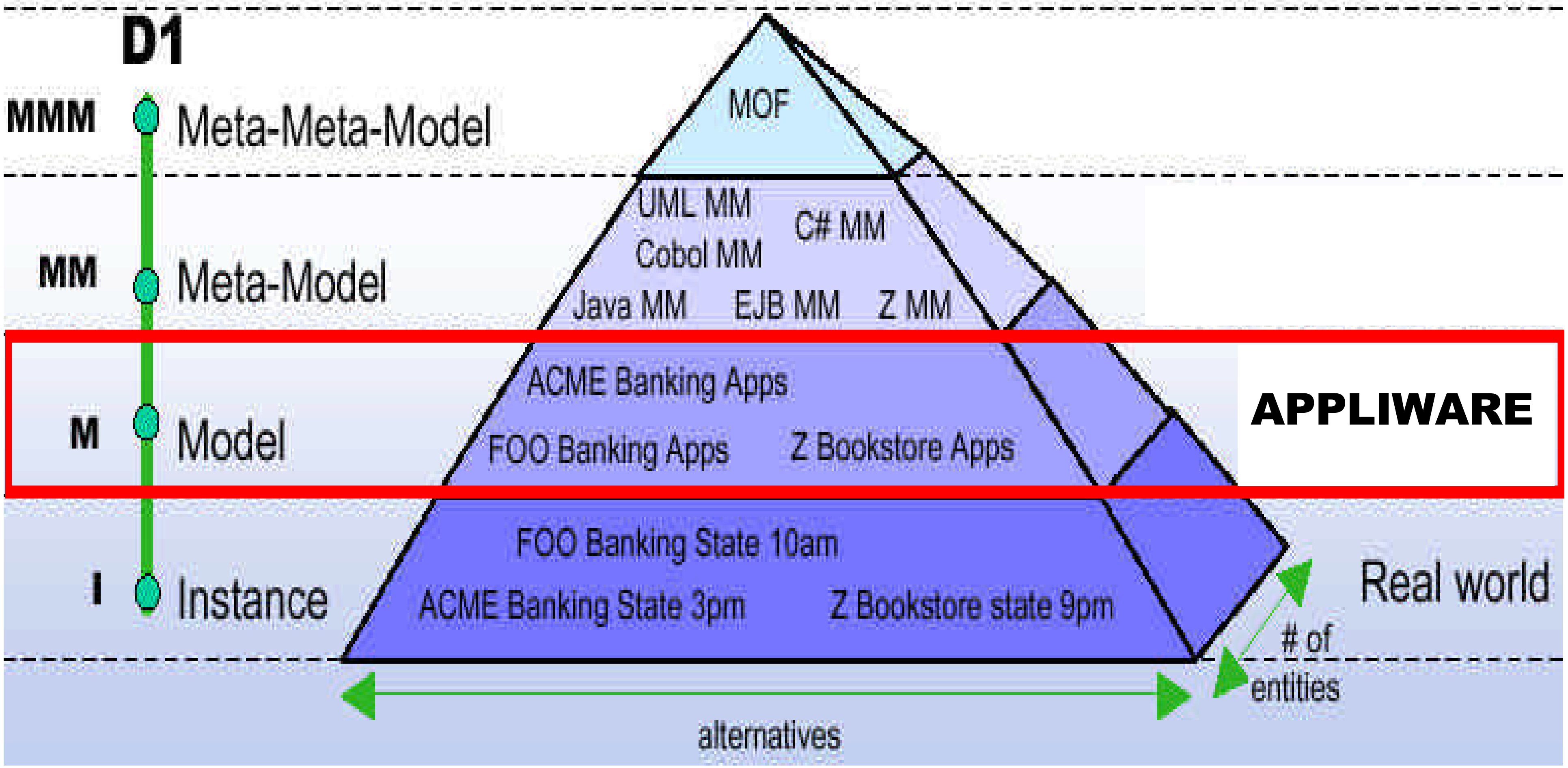
D1: The Meta-pyramid



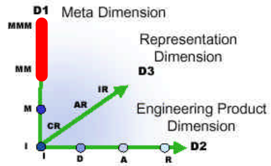


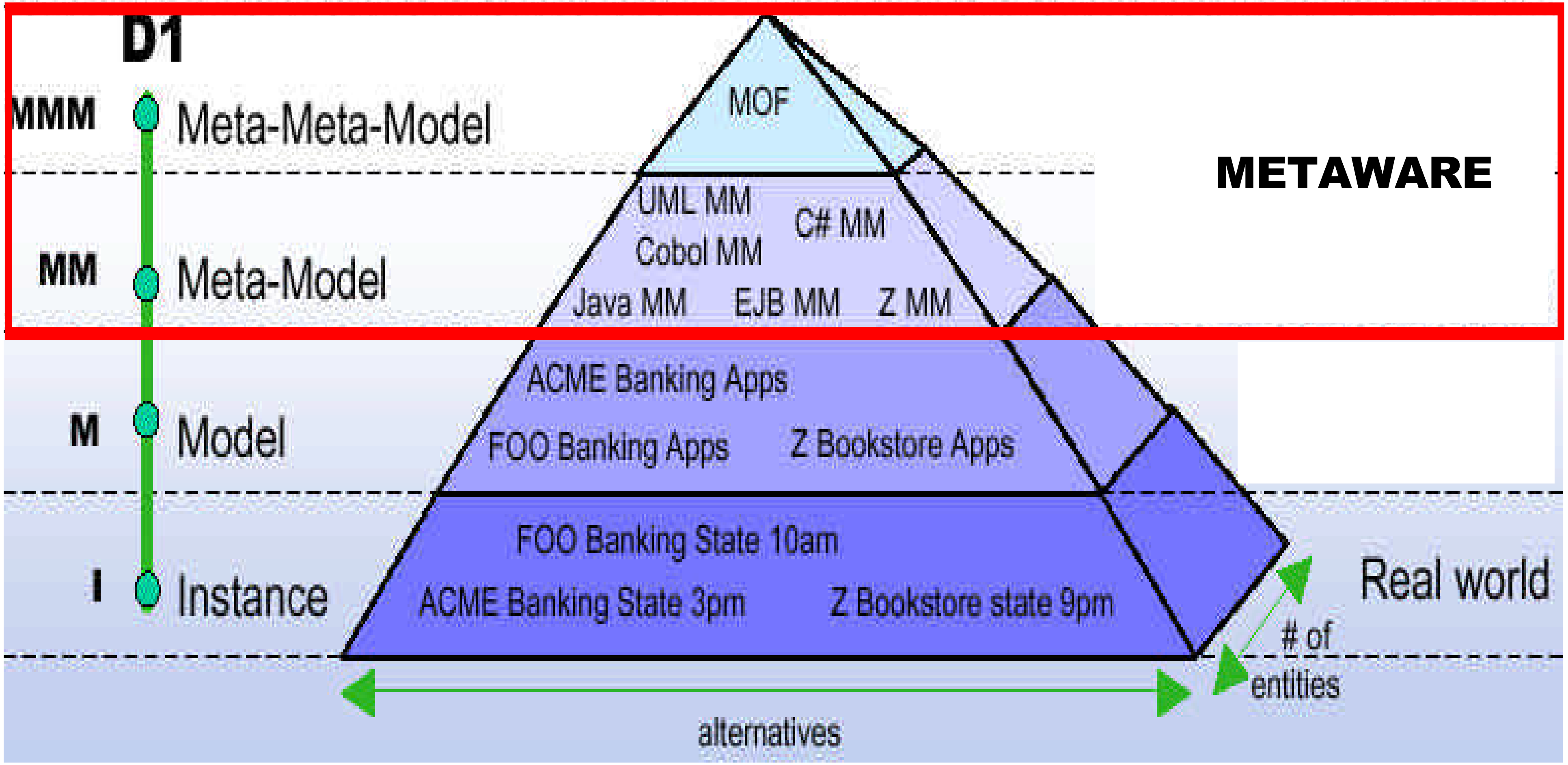
D1: The Meta-pyramid



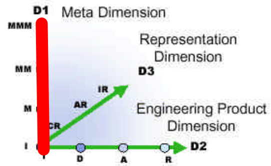


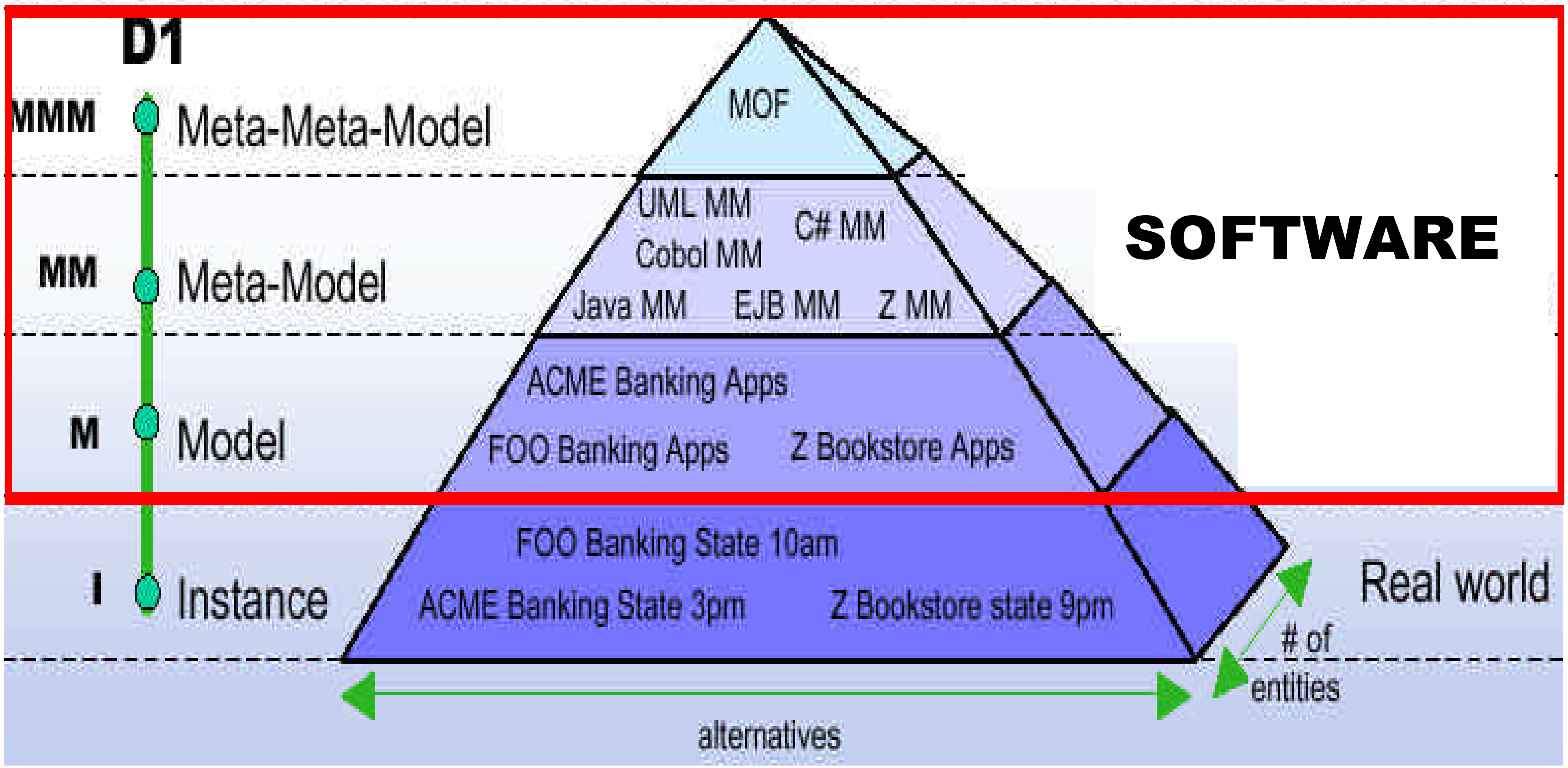
D1: The Meta-pyramid



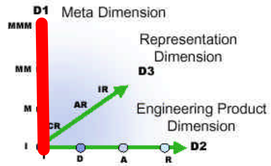


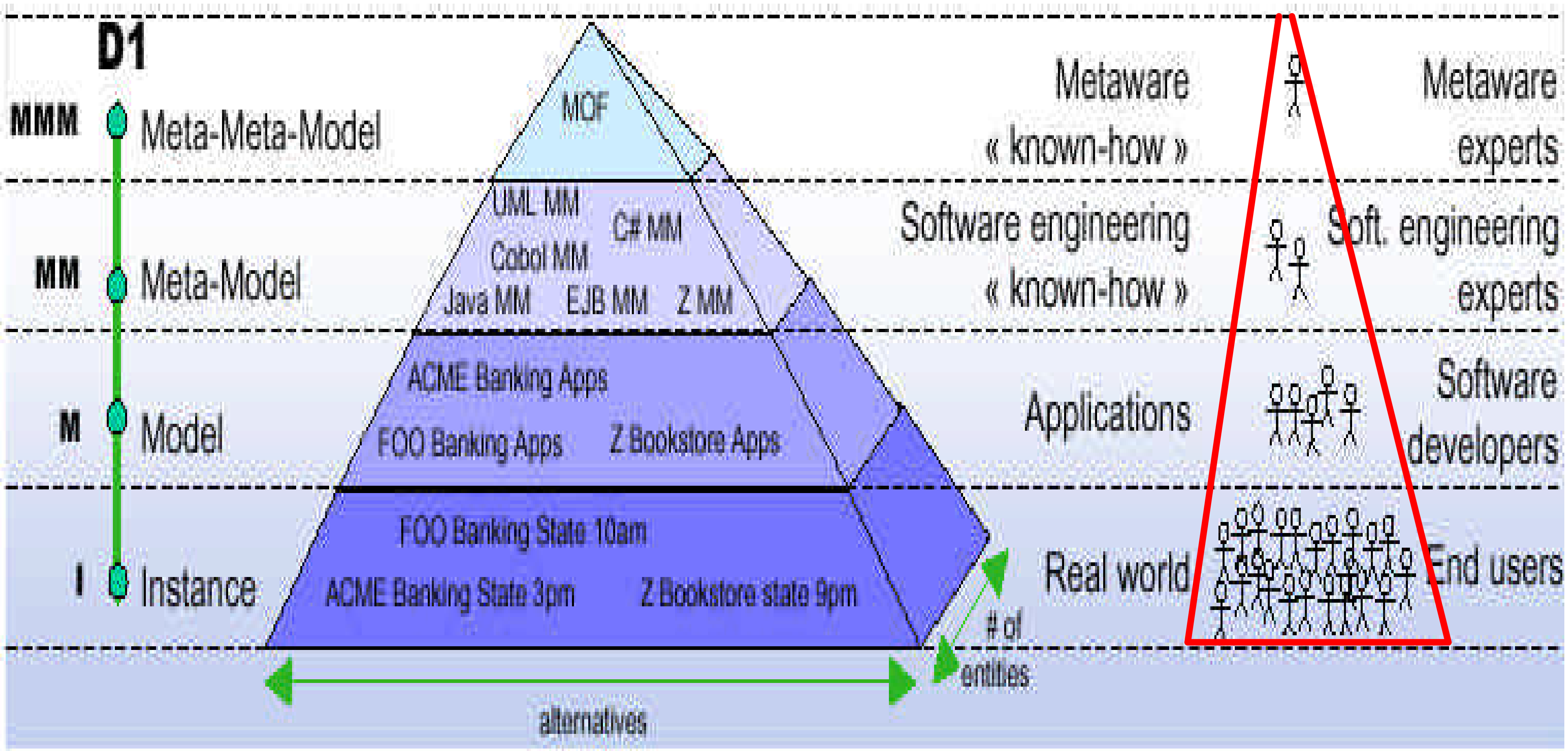
D1: The Meta-pyramid



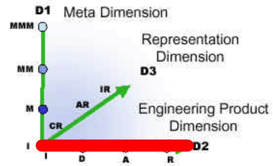


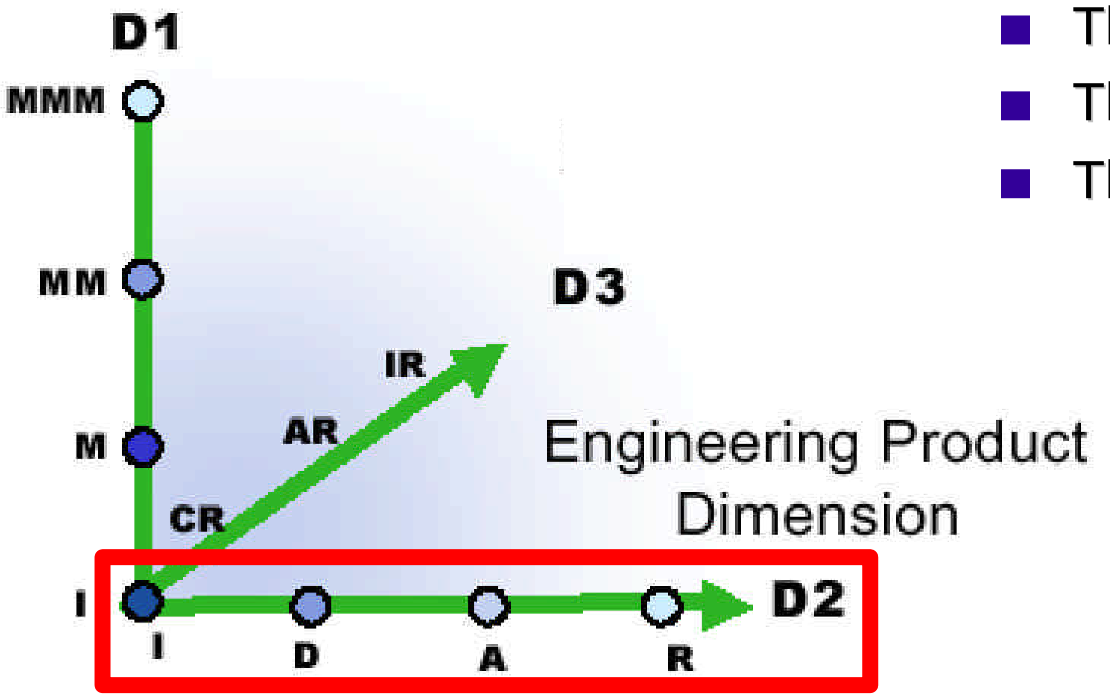
D1: The Meta actor pyramid



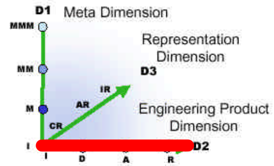


D2: The Engineering Dimension

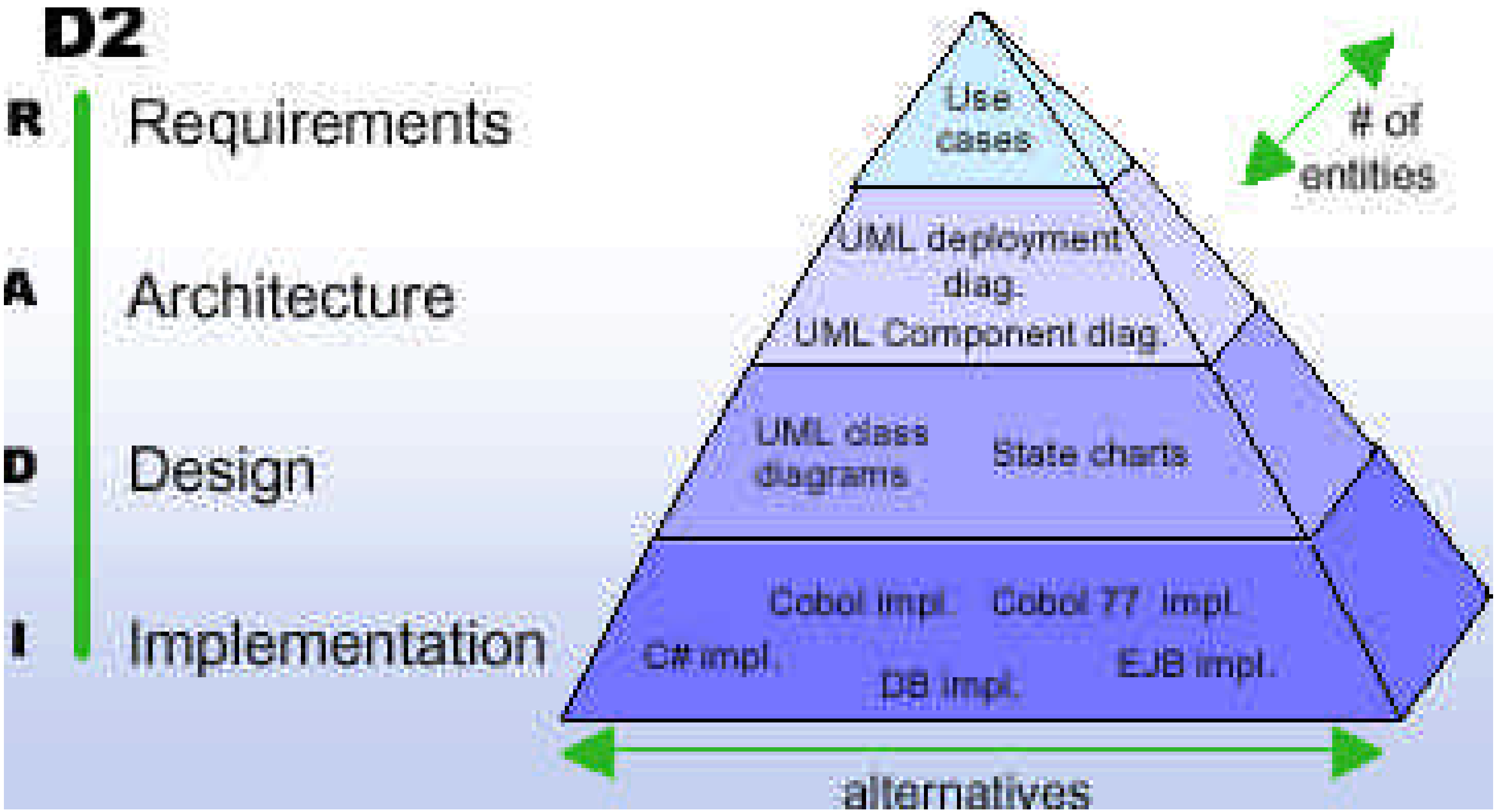




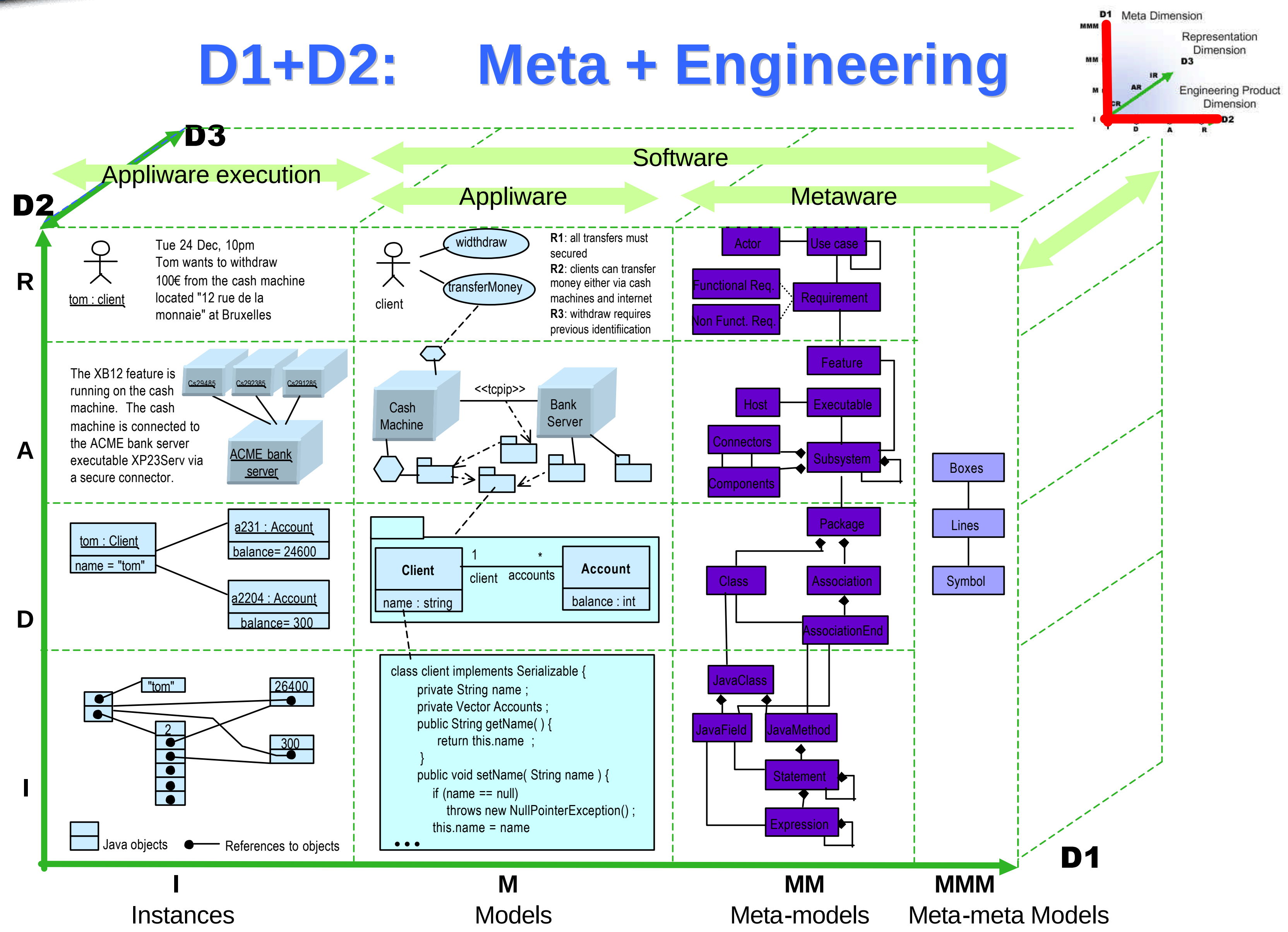
- The Engineering-tower
- The Engineering-pyramid
- The Engineering actor pyramid

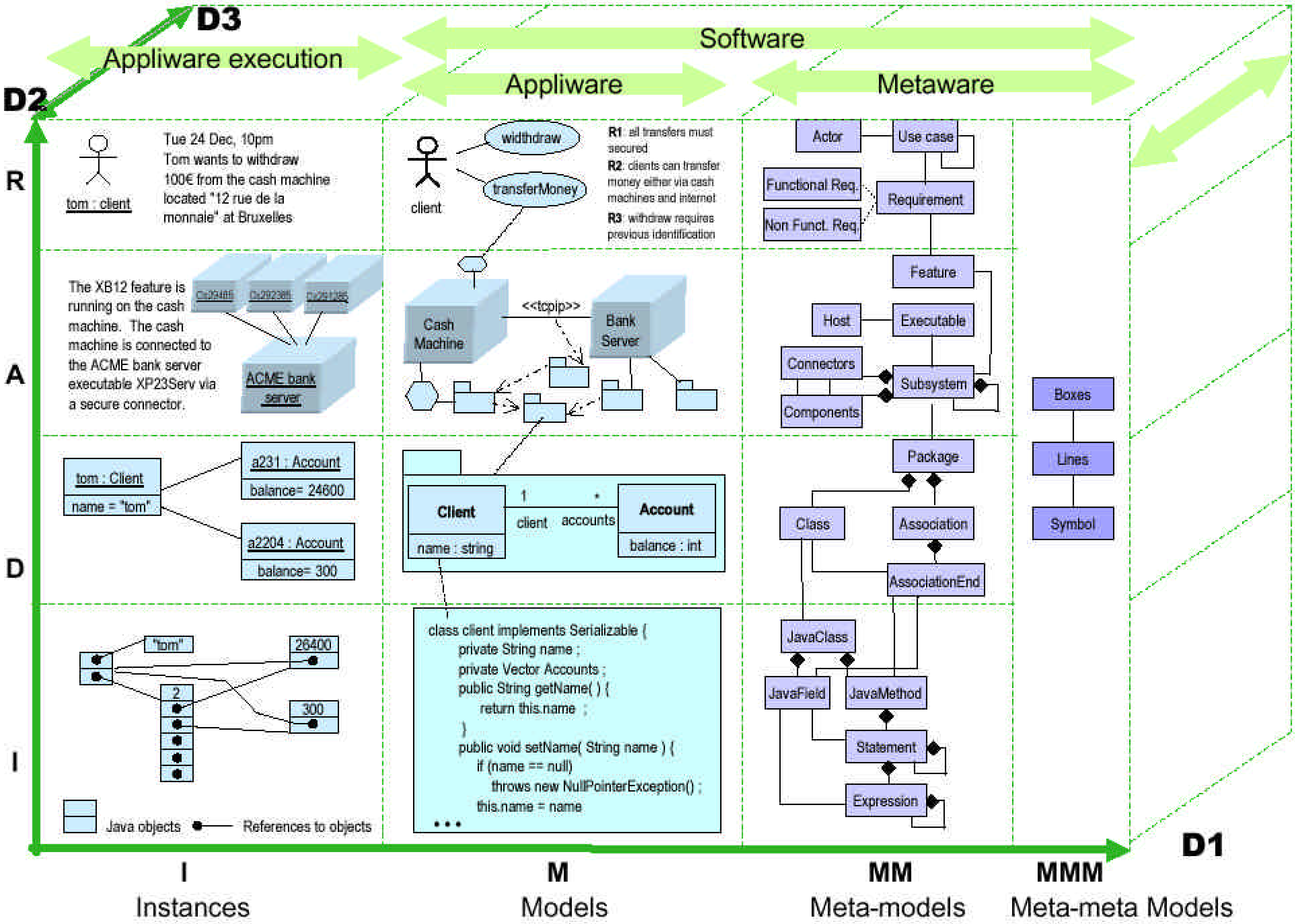


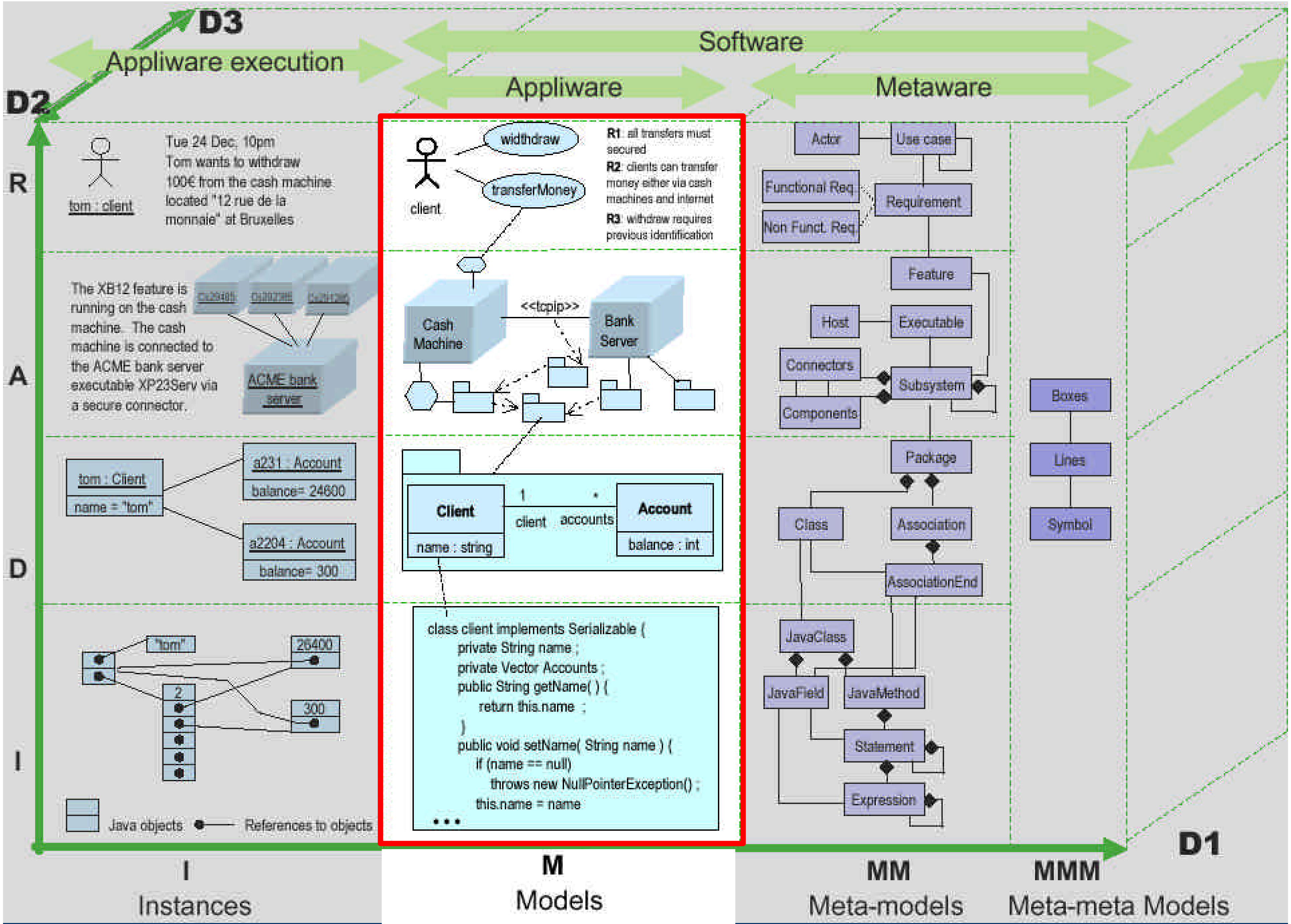
D2: The Engineering Pyramid

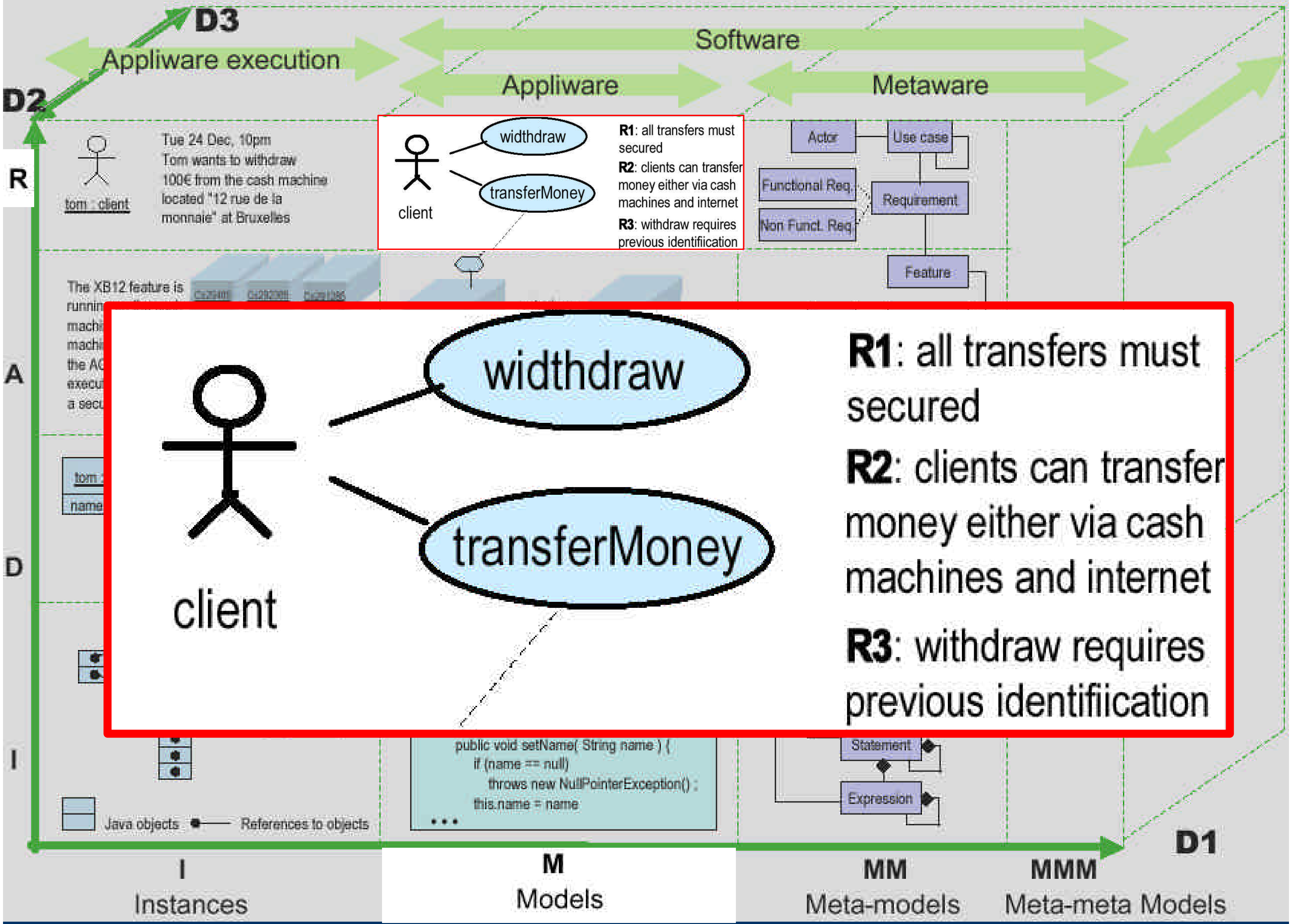


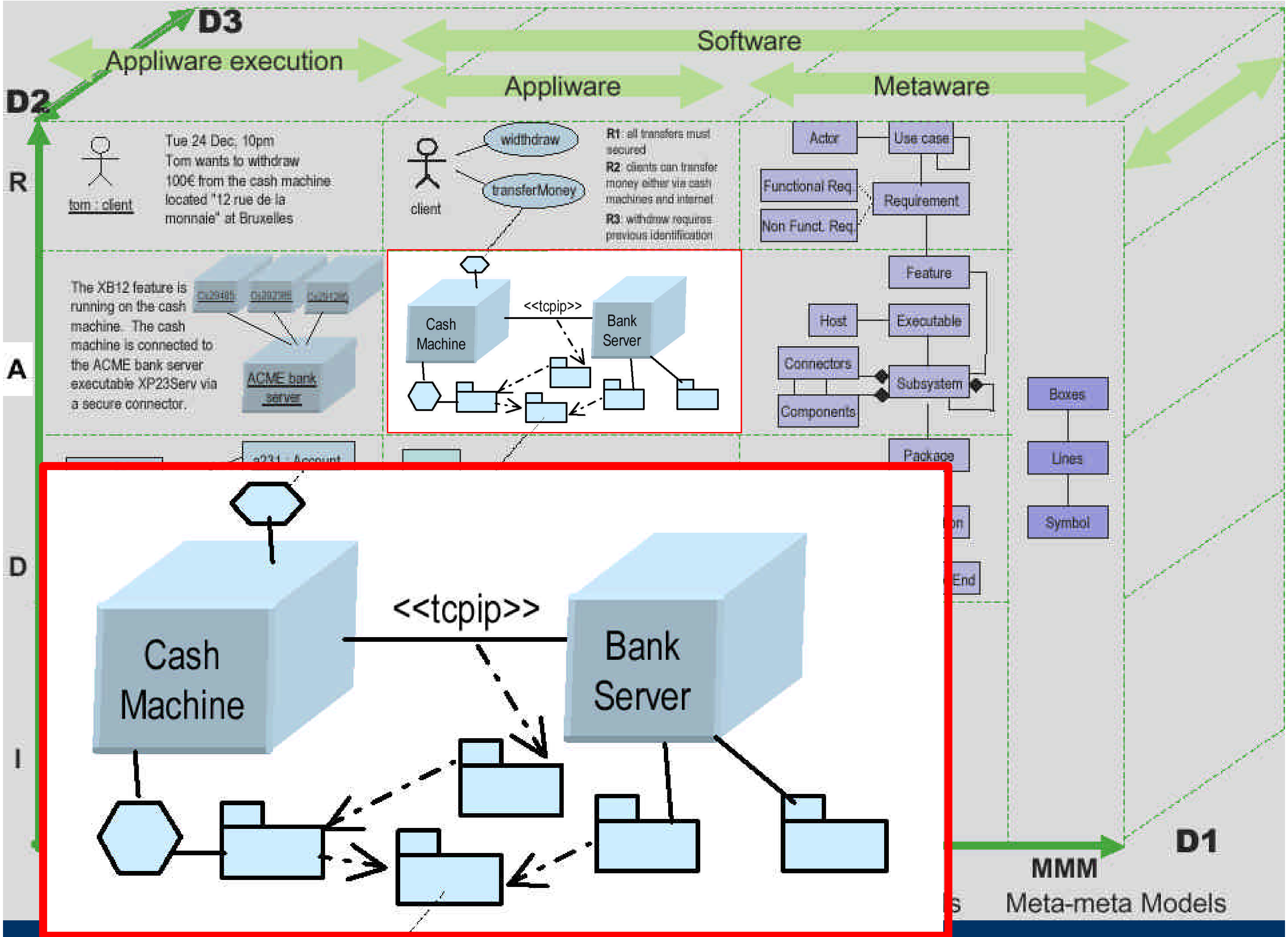
D1+D2: Meta + Engineering



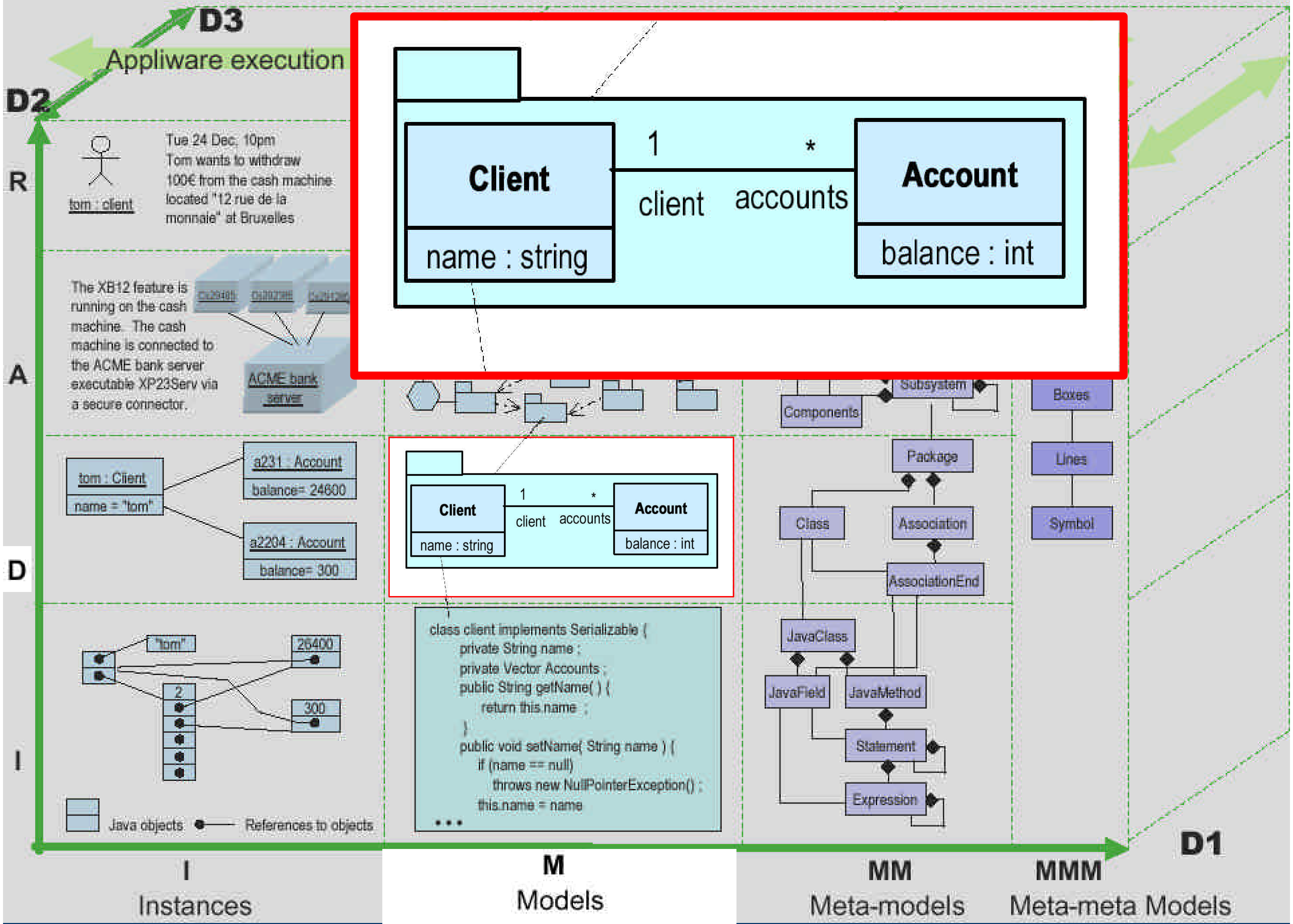

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnale" at Bruxelles
client
withdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

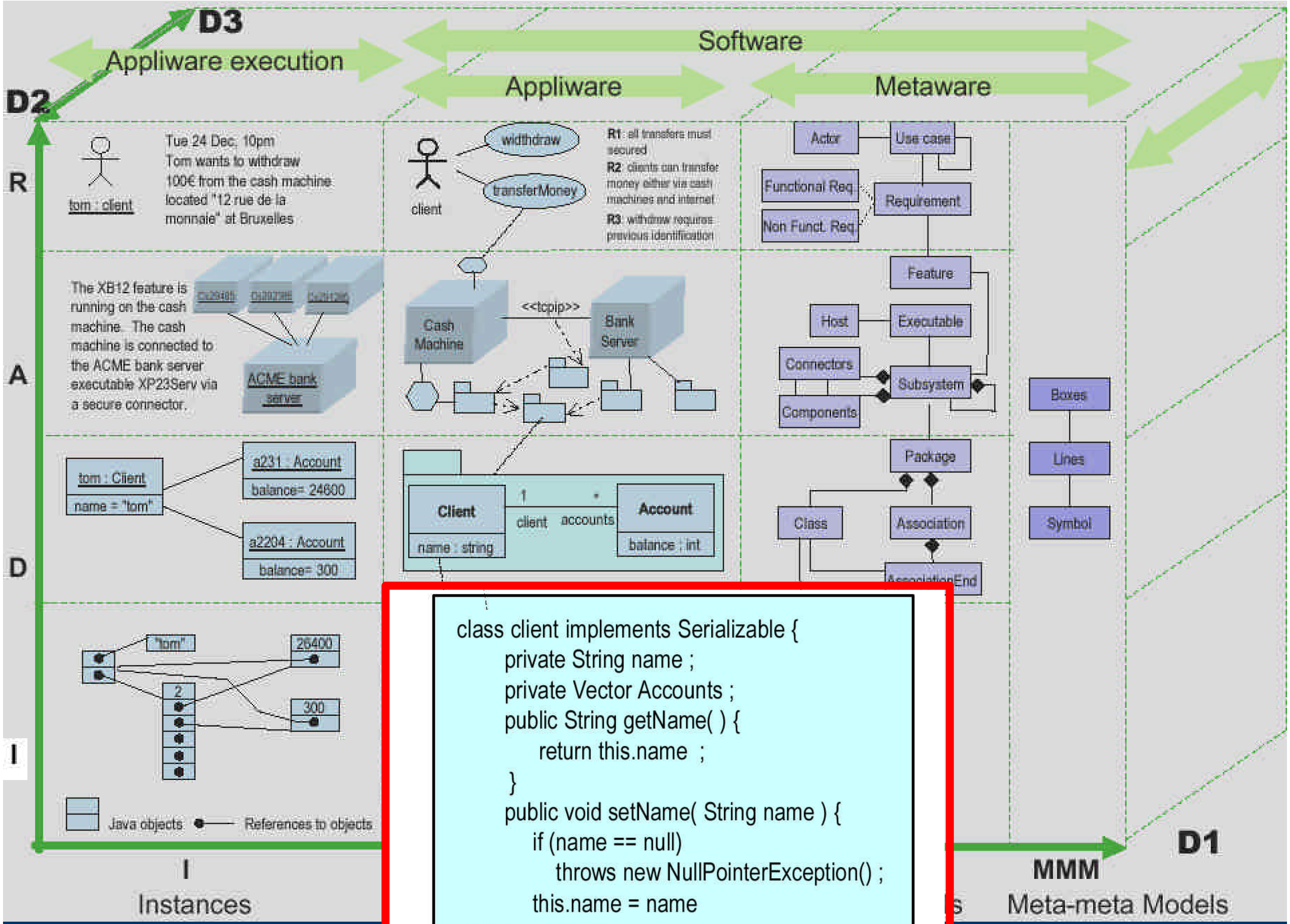

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec. 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
widthdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
<<tcpip>>
Cash Machine
Bank Server
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
• • •
"tom"
26400
2
300
Java objects
References to objects
Actor
Use case
Functional Req.
Non Funct. Req.
Requirement
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Class
Association
AssociationEnd
JavaClass
JavaField
JavaMethod
Statement
Expression
Boxes
Lines
Symbol
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models


D3
Appliware execution
Software
Appliware
Metaware
D2
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identifiication
Statement
Expression
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
Java objects
References to objects
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

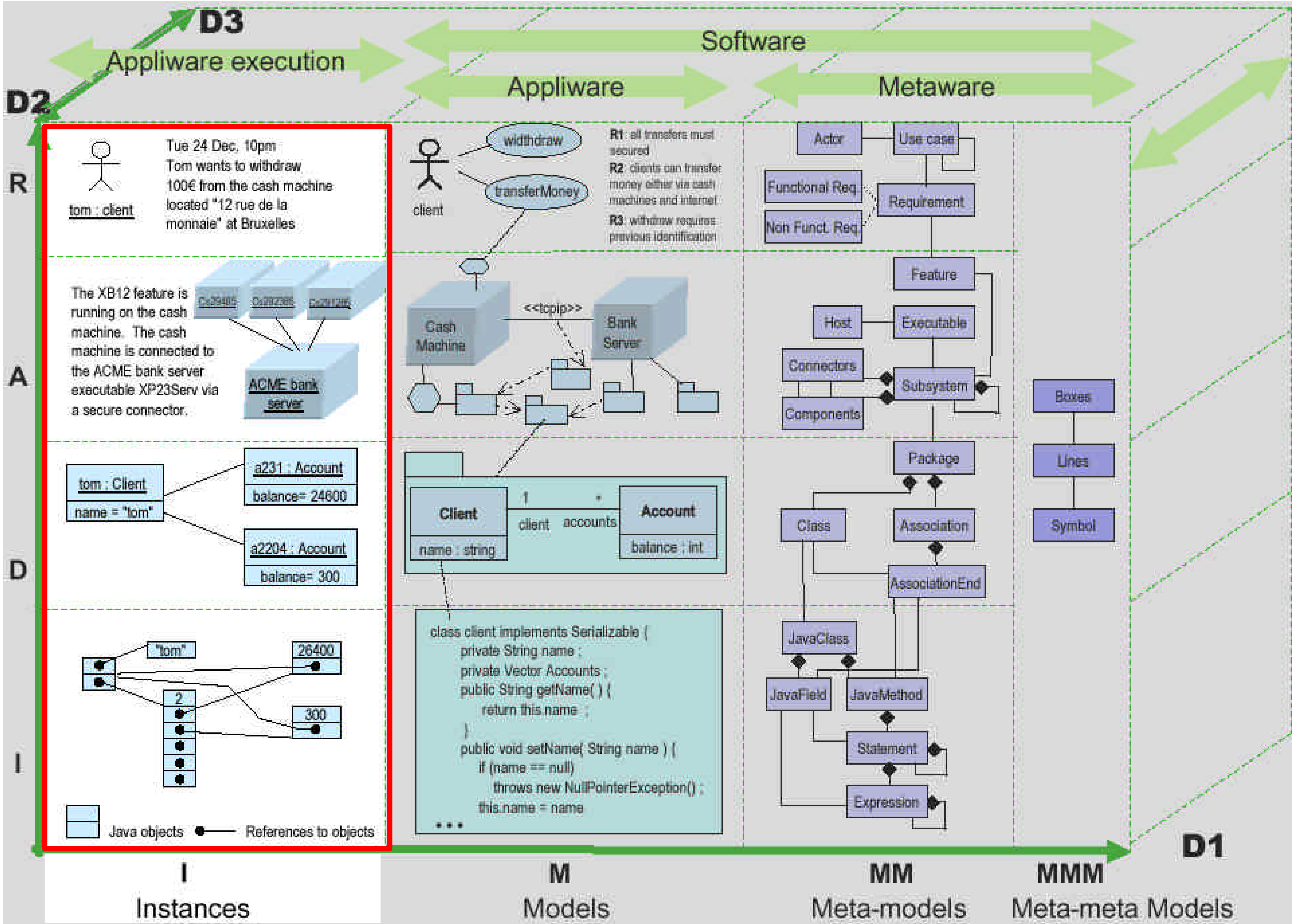


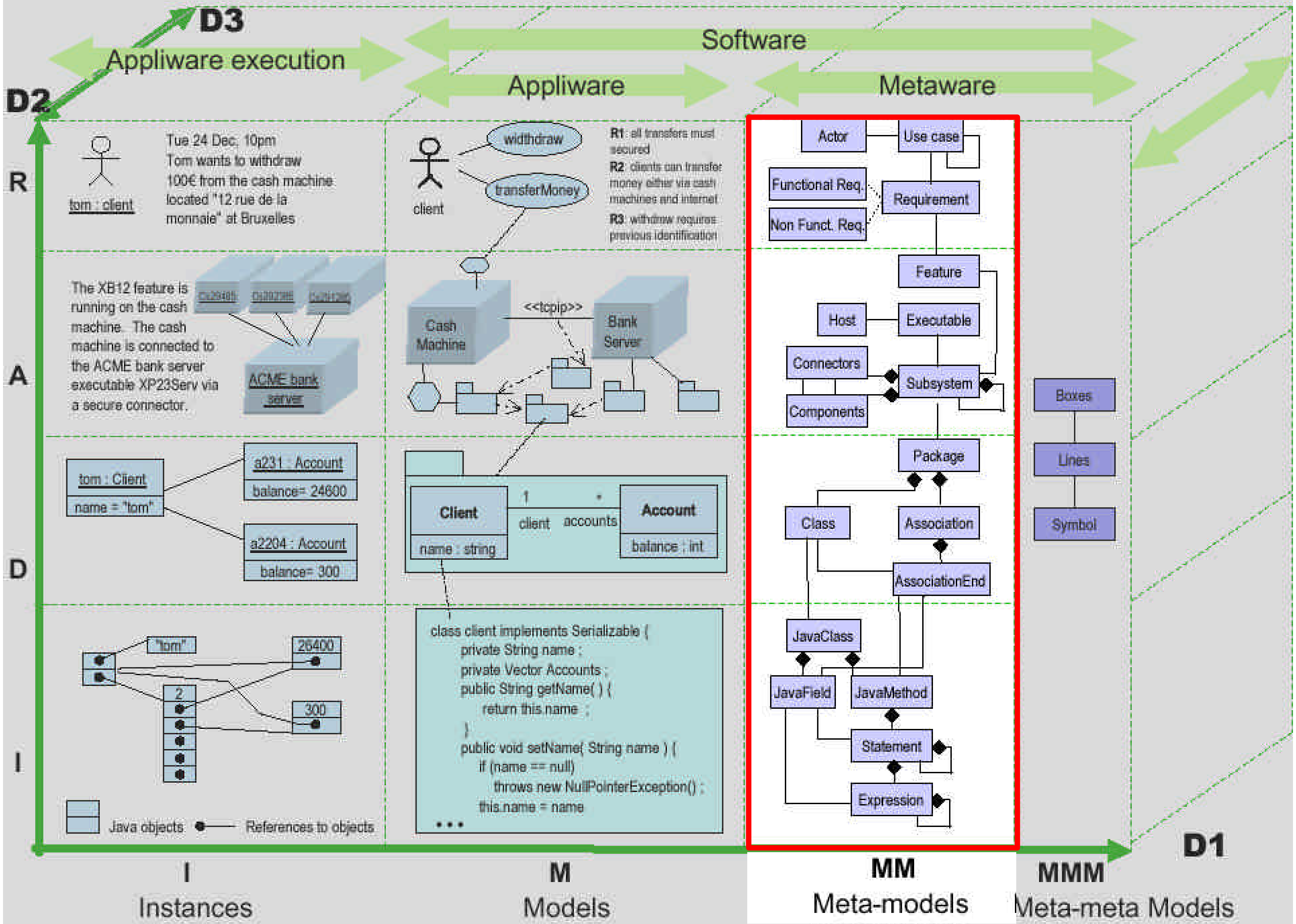
D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Boxes
Lines
Symbol
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
a231 : Account
Cash Machine
Bank Server
<<tcpip>>
MMM
Meta-meta Models


D3
Appliware execution
D2
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
The XB12 feature is
running on the cash
machine. The cash
machine is connected to
the ACME bank server
executable XP23Serv via
a secure connector.
ACME bank
server
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
"tom"
26400
2
300
Java objects
References to objects
Client
name : string
1
*
client
accounts
Account
balance : int
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
Subsystem
Components
Package
Class
Association
AssociationEnd
JavaClass
JavaField
JavaMethod
Statement
Expression
Boxes
Lines
Symbol
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

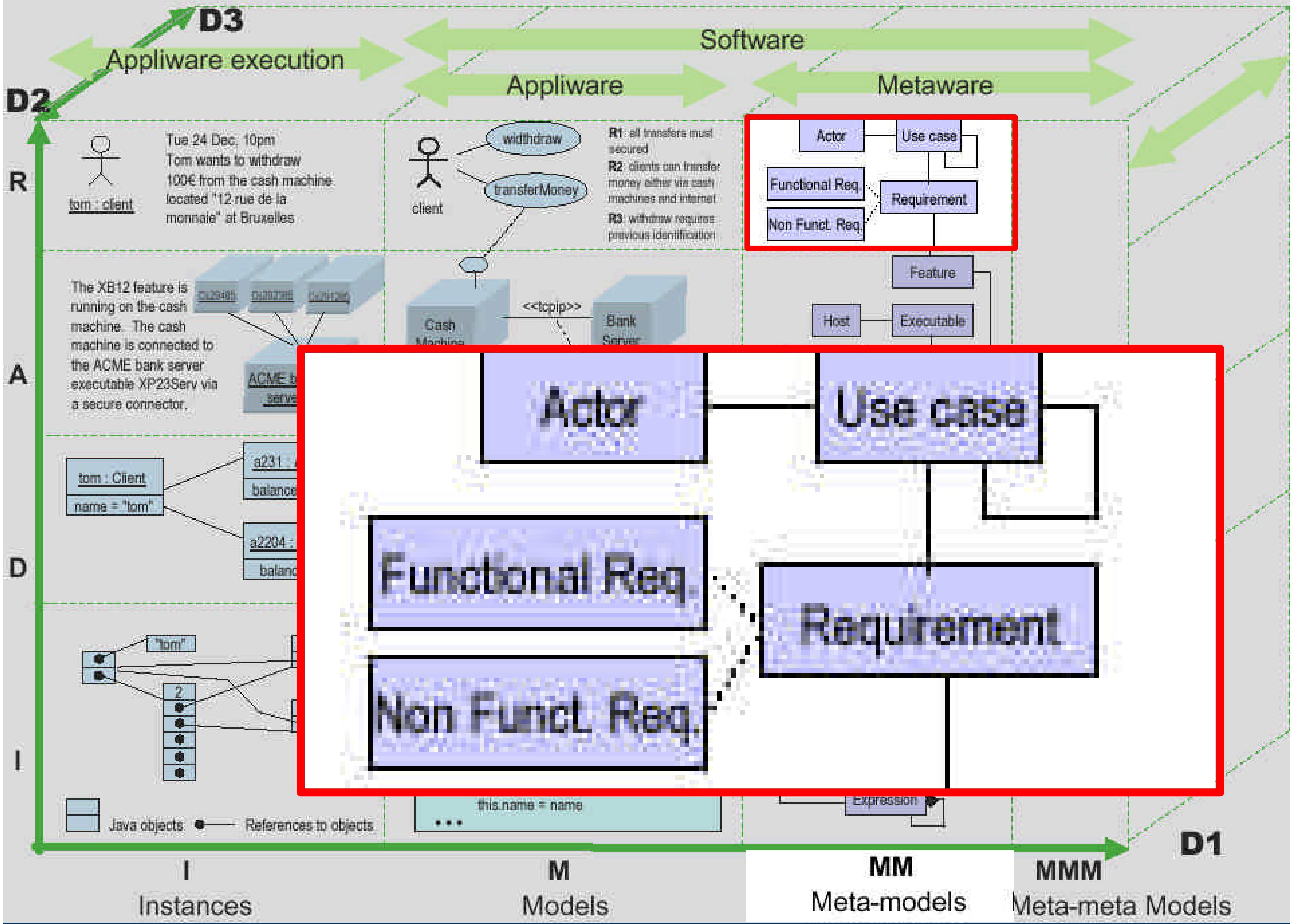



```
class client implements Serializable {
    private String name ;
    private Vector Accounts ;
    public String getName( ) {
        return this.name ;
    }
    public void setName( String name ) {
        if (name == null)
            throws new NullPointerException() ;
        this.name = name
    • • •
```

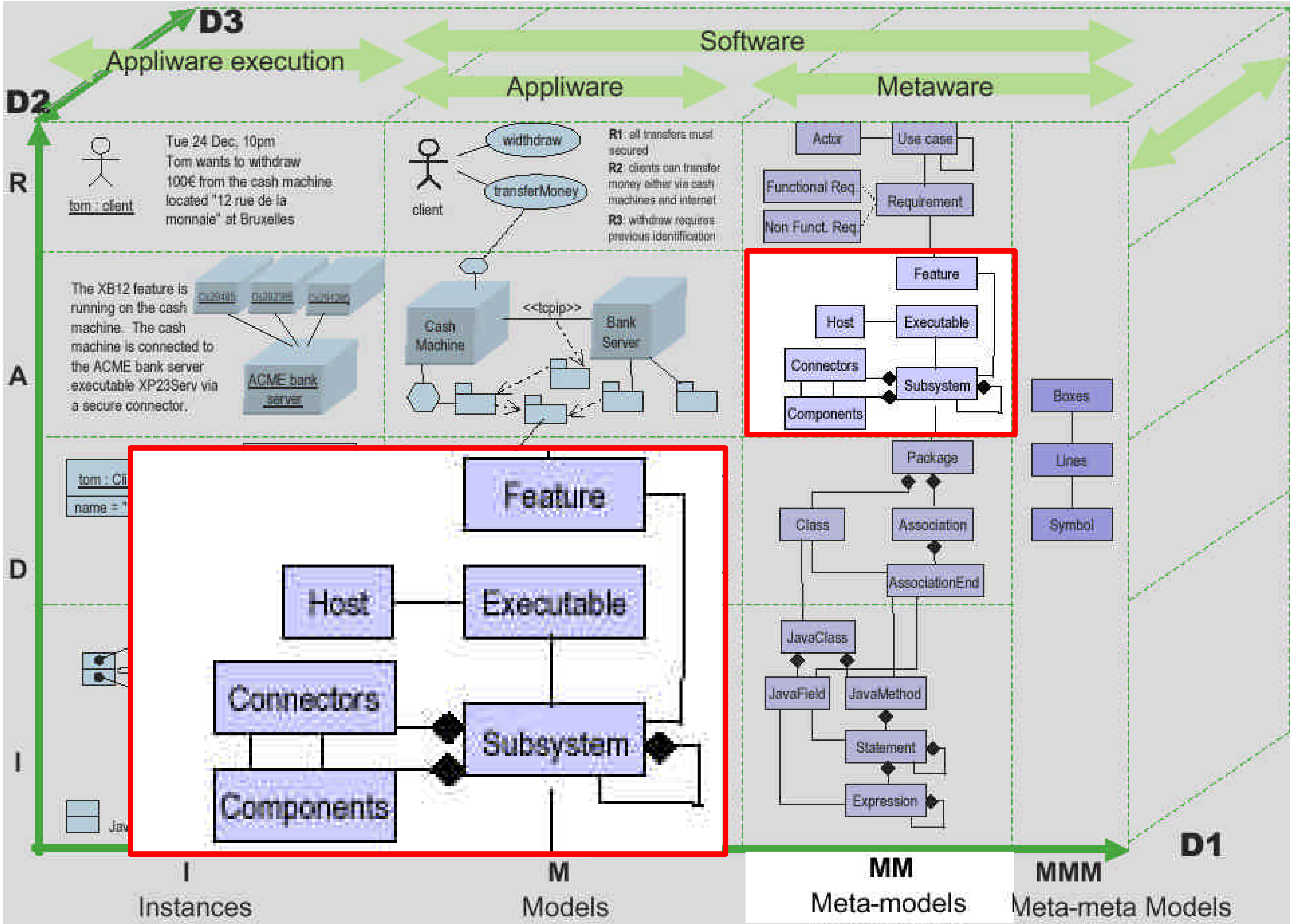

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
"tom"
26400
300
2
Java objects
References to objects
widthdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Cash Machine
<<tcpip>>
Bank Server
Client
name : string
1
client
*
accounts
Account
balance : int
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Class
Association
AssociationEnd
JavaClass
JavaField
JavaMethod
Statement
Expression
Boxes
Lines
Symbol
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

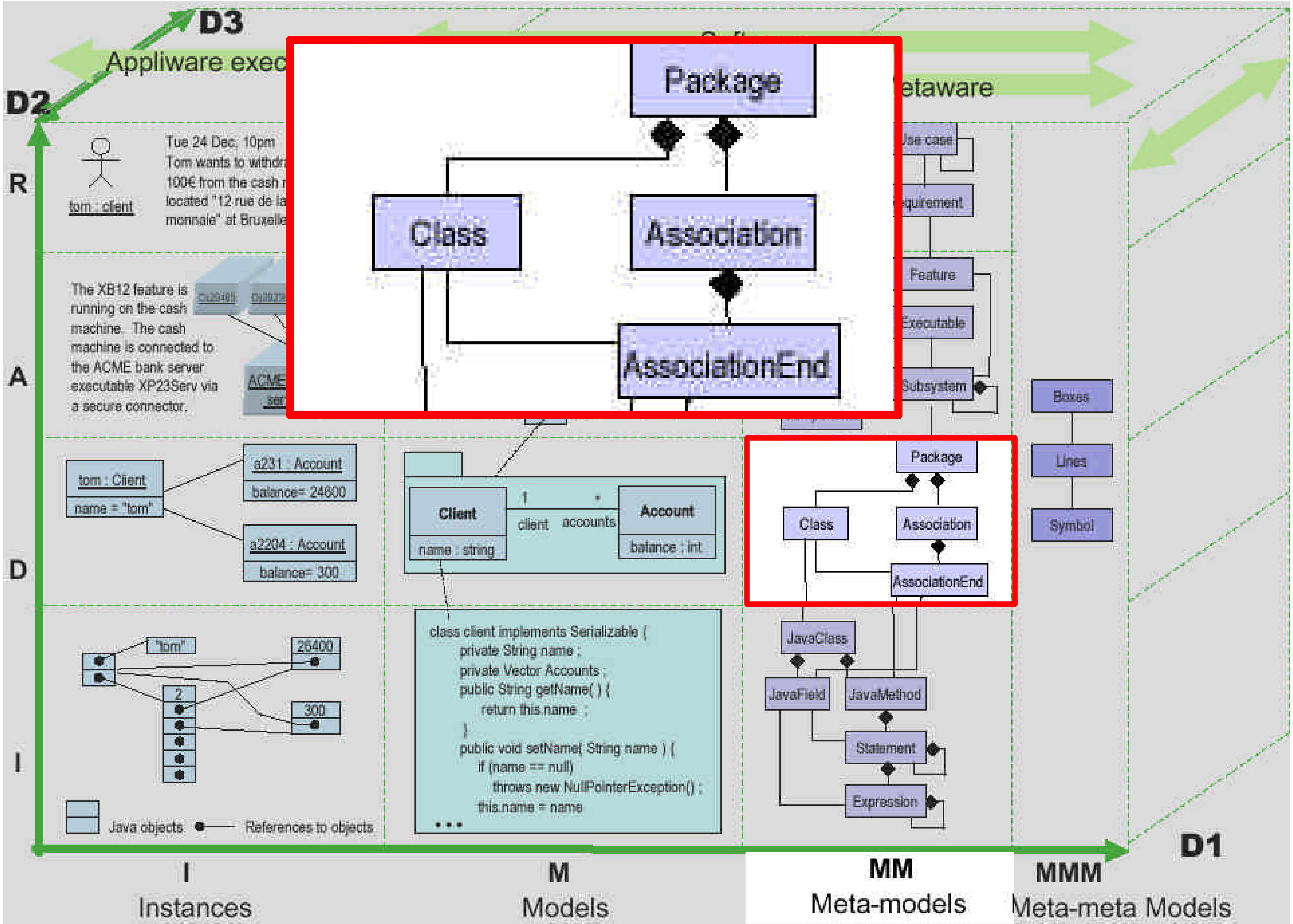


D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
tom : client
widthdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Class
Association
AssociationEnd
JavaClass
JavaField
JavaMethod
Statement
Expression
Boxes
Lines
Symbol
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

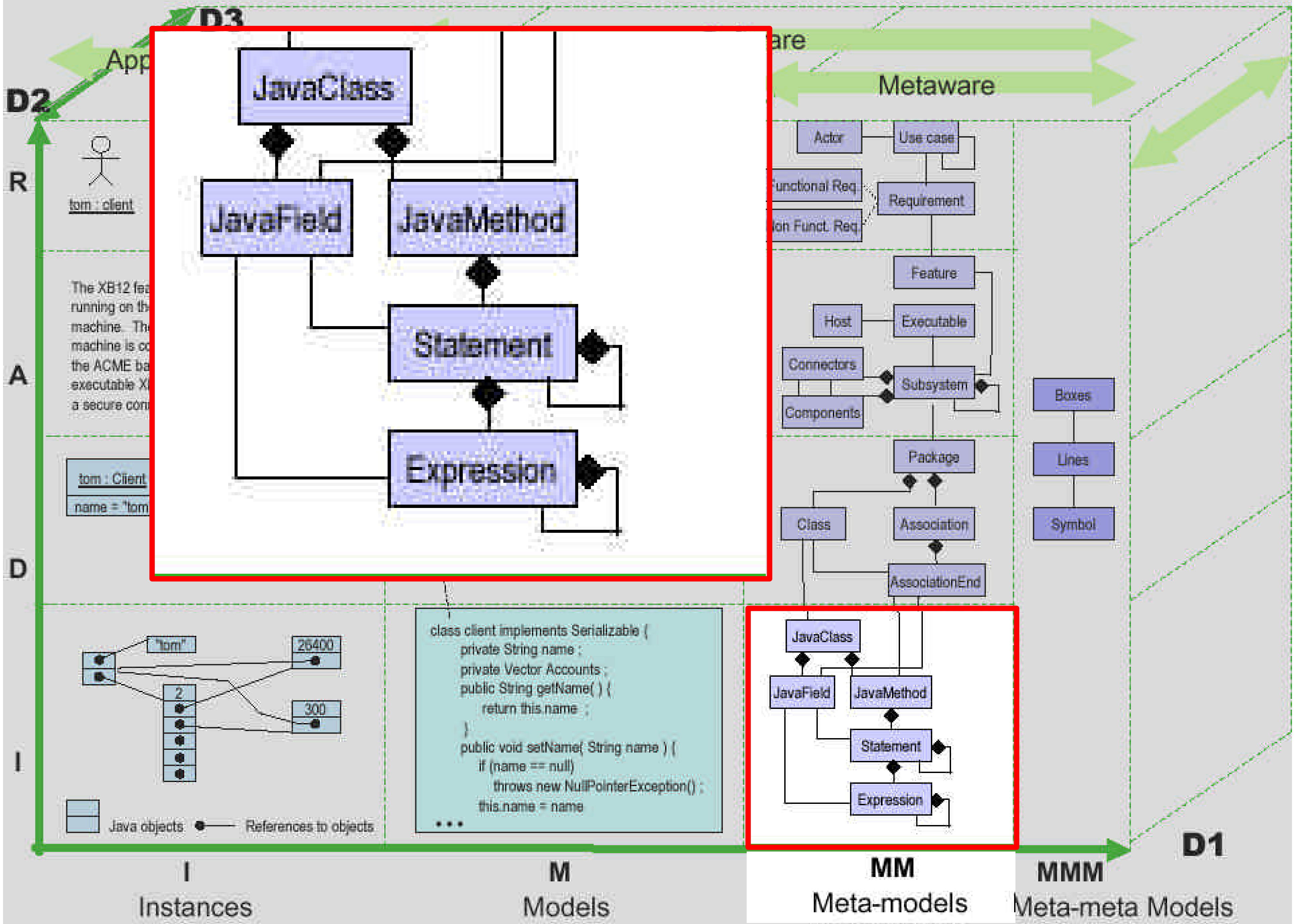


D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
tom : client
widthdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
Host
Executable
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
<<tcpip>>
Cash
Bank Server
tom : Client
name = "tom"
"tom"
2
Java objects
References to objects
this.name = name
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

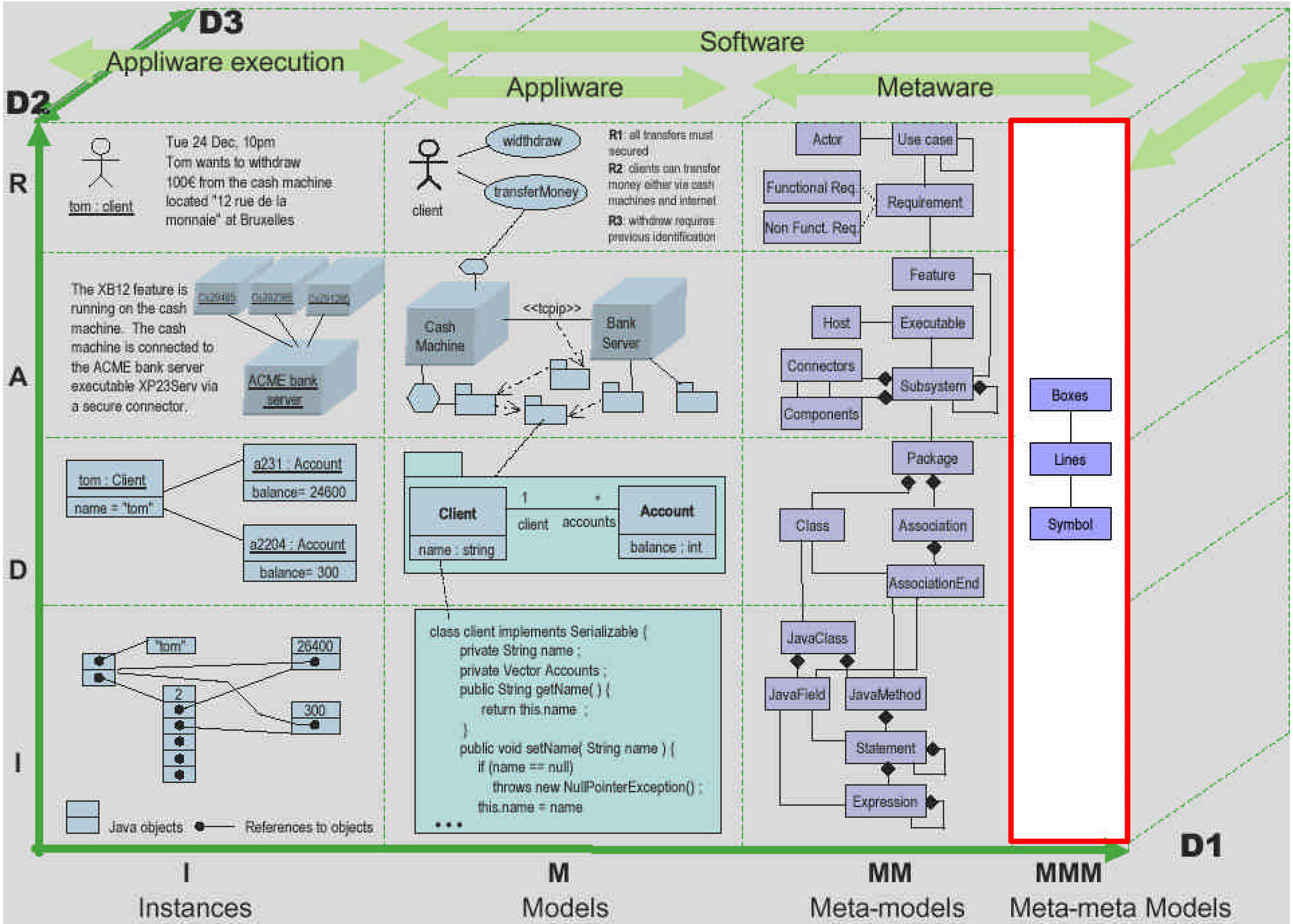

D3
Appliware execution
Software
Appliware
Metaware
D2
R
A
D
I
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
tom : client
client
withdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
Package
Class
Association
AssociationEnd
JavaClass
JavaField
JavaMethod
Statement
Expression
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

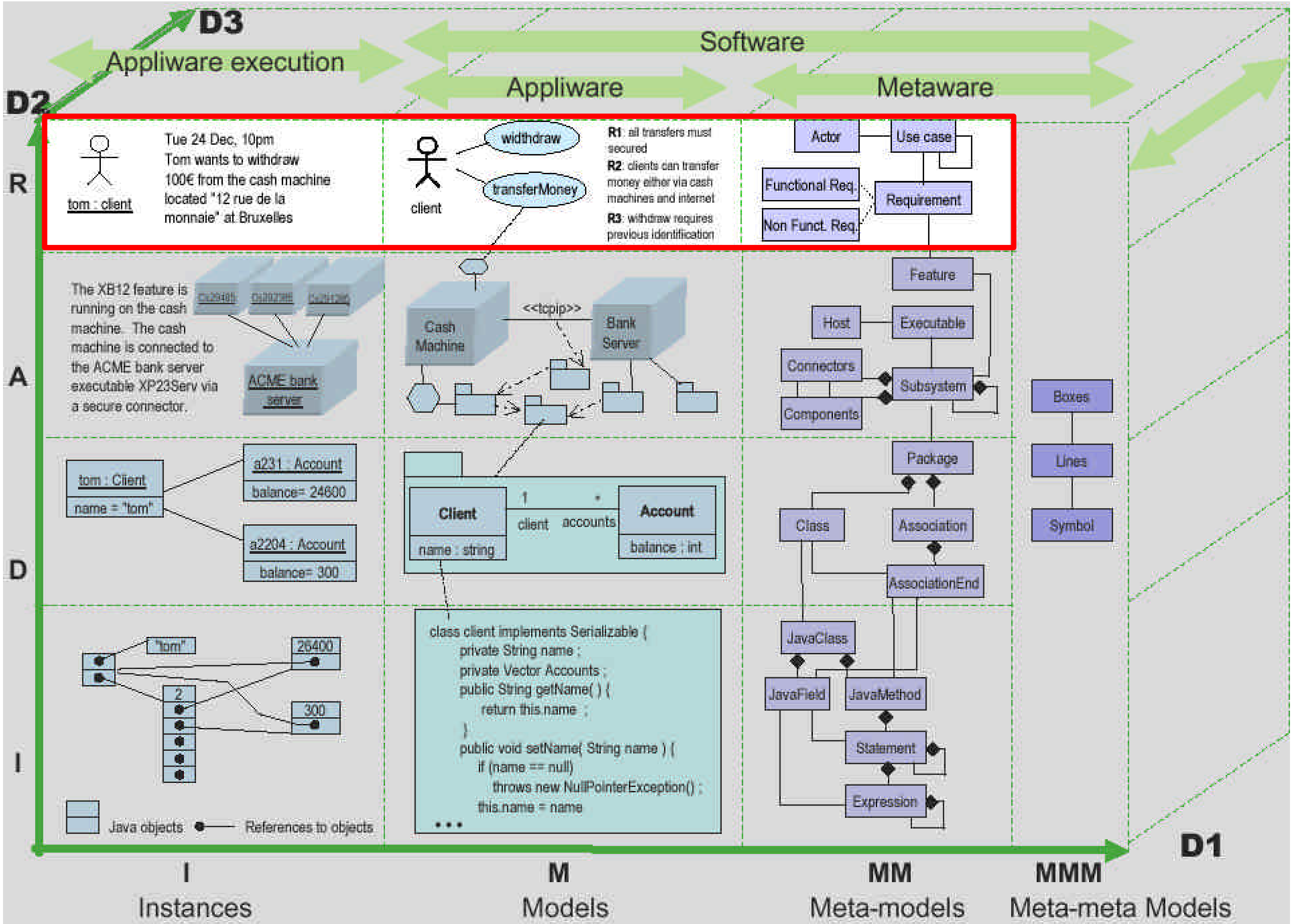


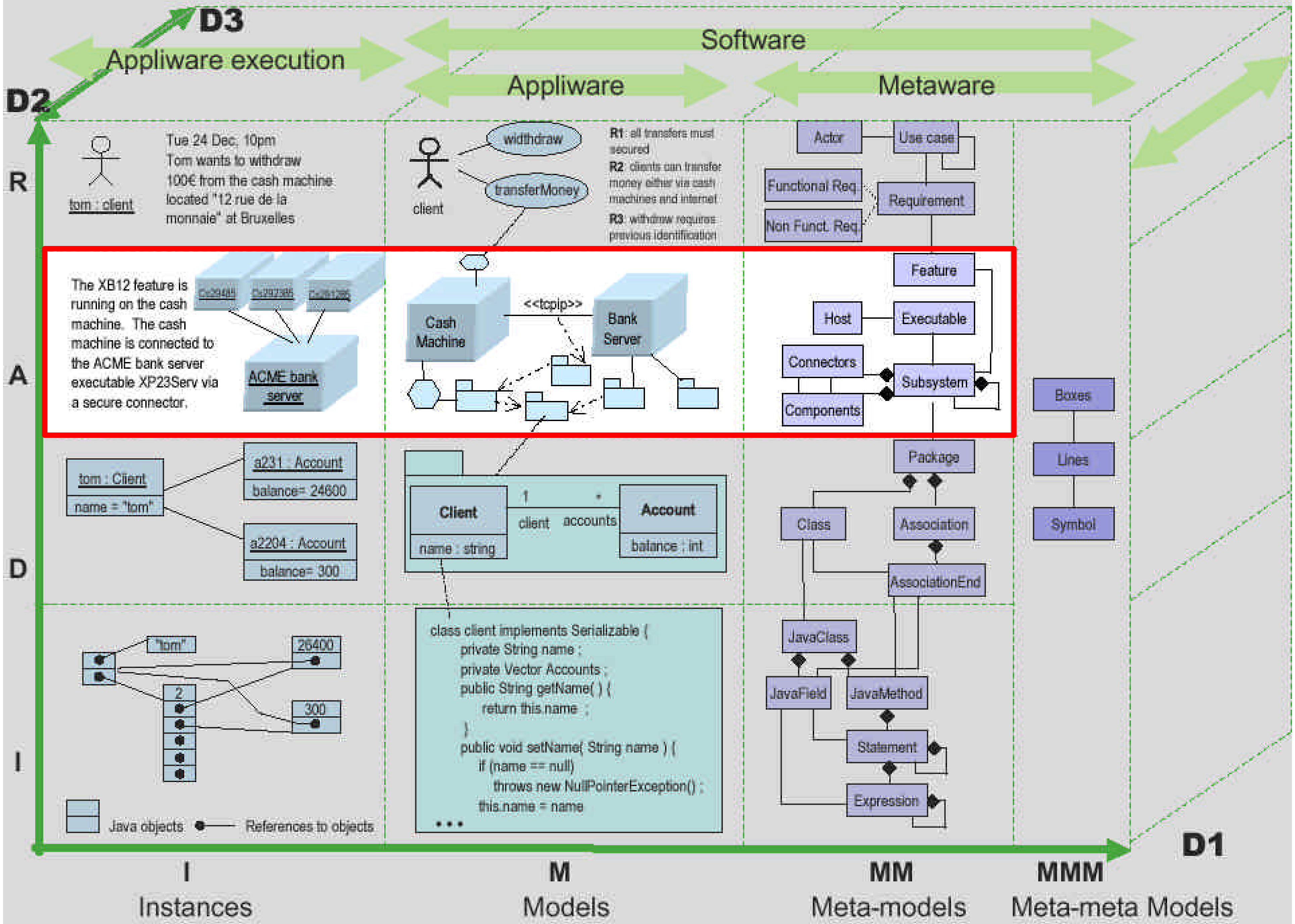
D3
D2
D1
Appliware exec
etaware
Package
Class
Association
AssociationEnd
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdr
100€ from the cash
located "12 rue de la
monnaie" at Bruxelle
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
*
client
accounts
Account
balance : int
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
Use case
Requirement
Feature
Executable
Subsystem
JavaClass
JavaField
JavaMethod
Statement
Expression
Boxes
Lines
Symbol
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

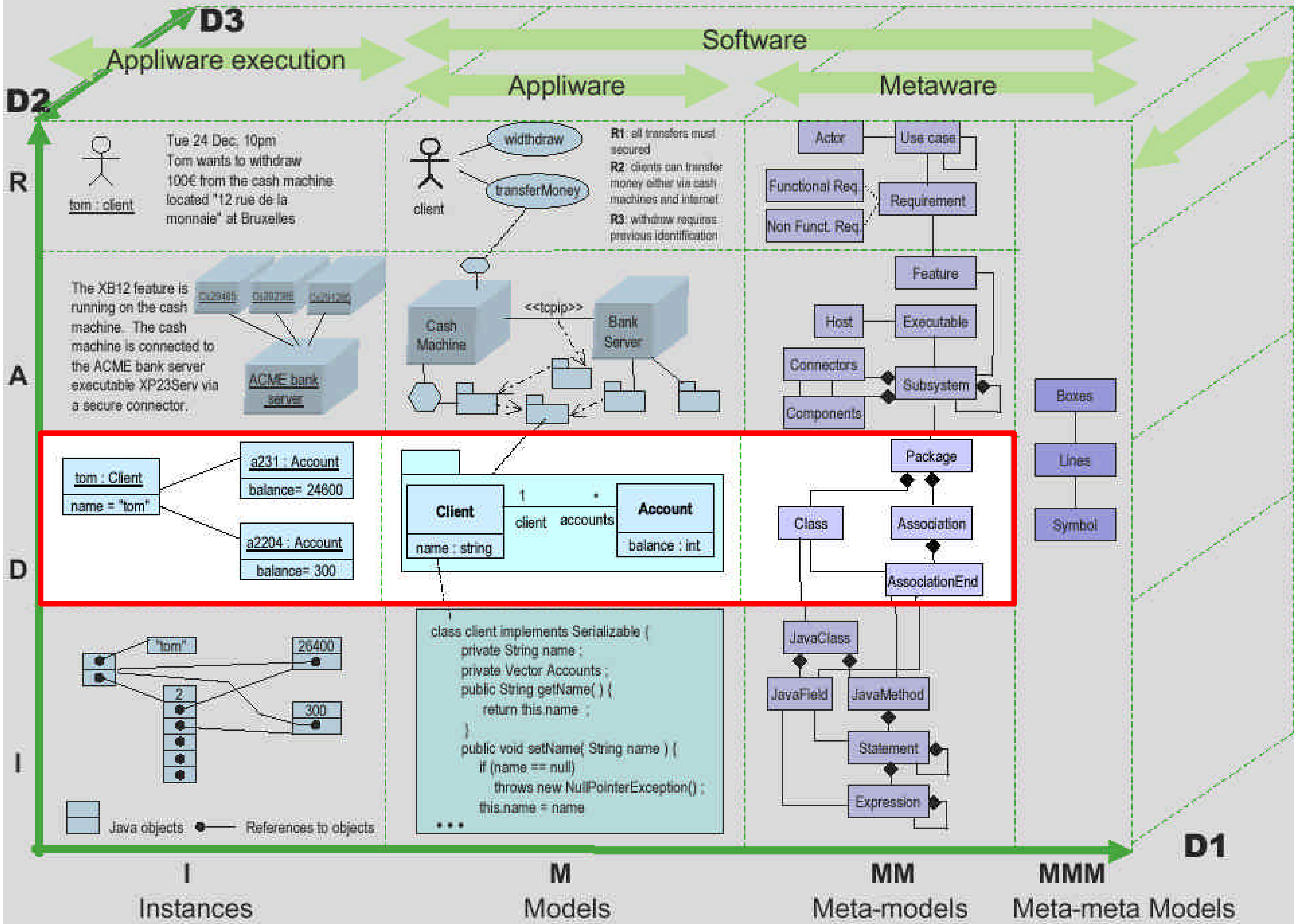


D3
D2
D1
App
are
Metaware
R
A
D
I
tom : client
The XB12 fea
running on th
machine. Th
machine is co
the ACME ba
executable X
a secure con
tom : Client
name = "tom
JavaClass
JavaField
JavaMethod
Statement
Expression
Actor
Use case
Functional Req.
Non Funct. Req.
Requirement
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Class
Association
AssociationEnd
Boxes
Lines
Symbol
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

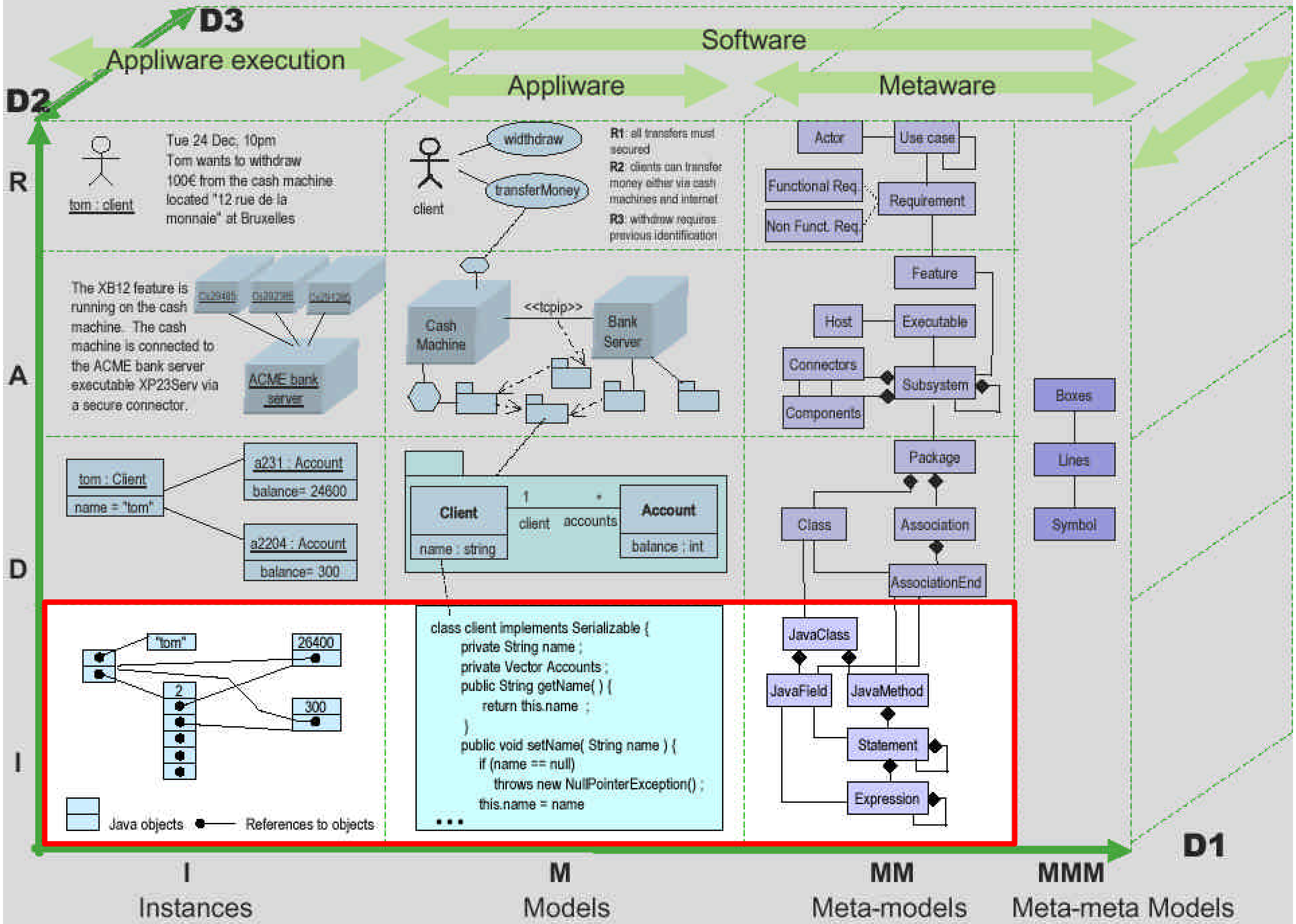

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
tom : client
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

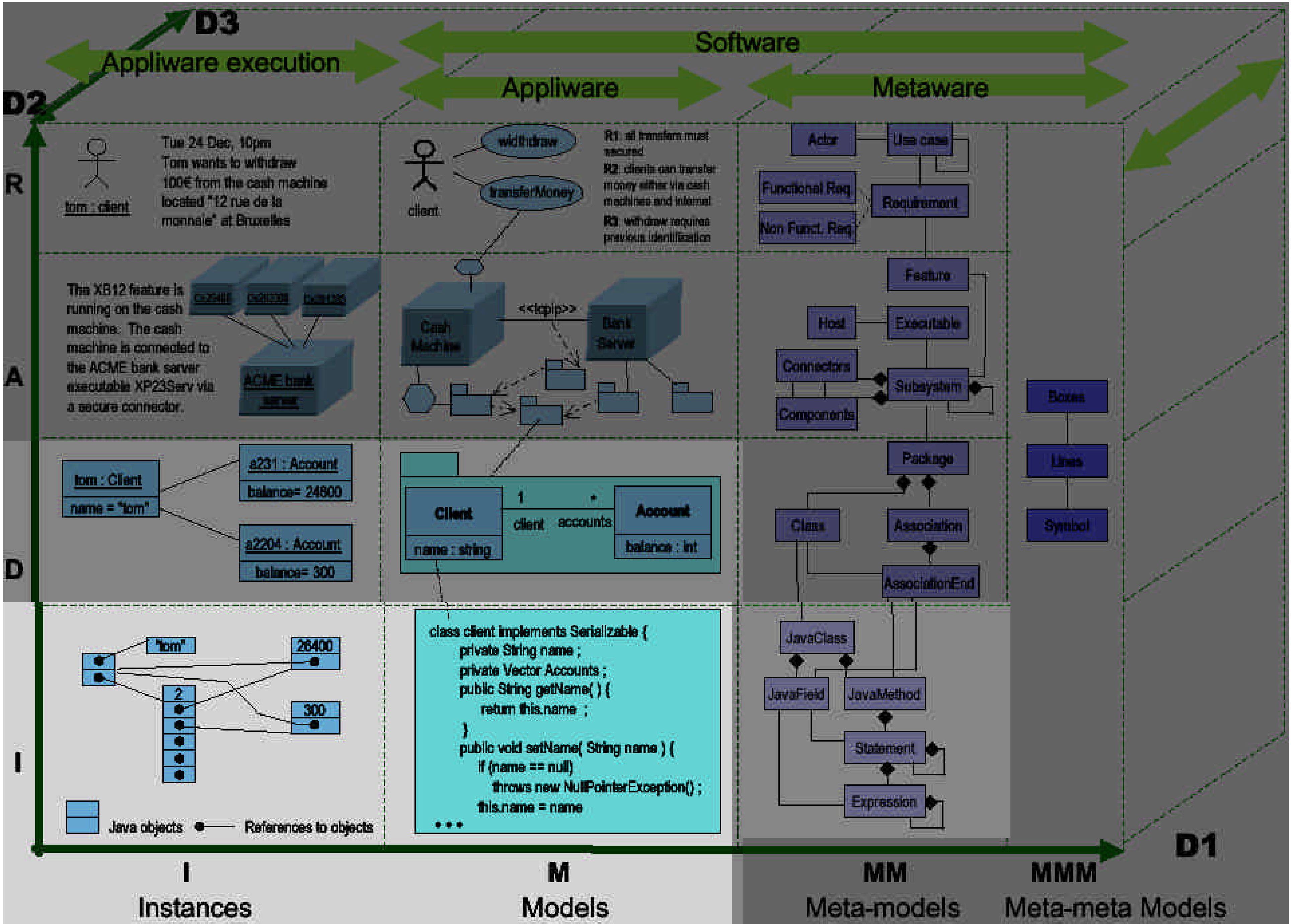

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
withdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
Lines
Symbol
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
• • •
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

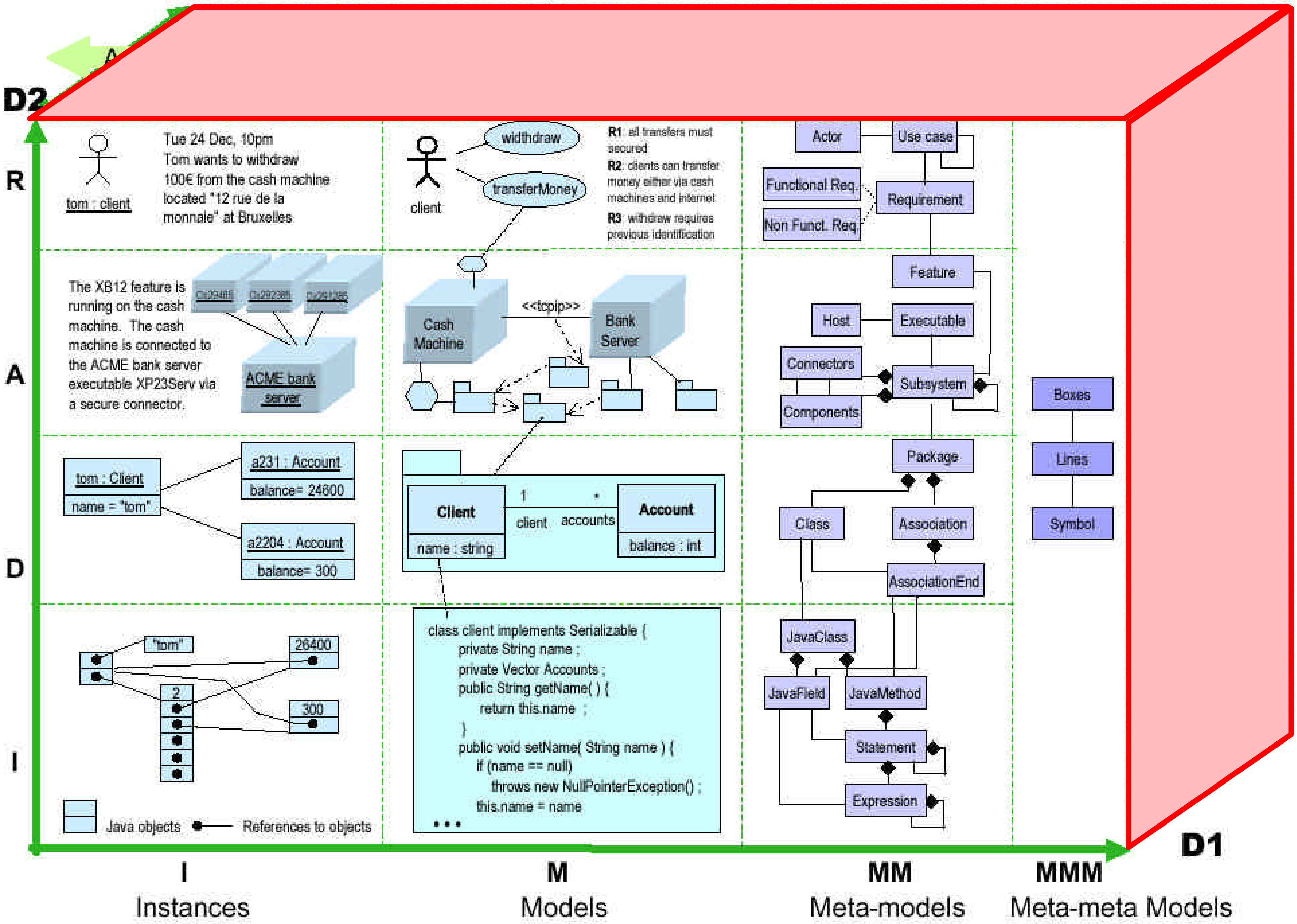

D3
Appliware execution
Software
Appliware
Metaware
D2
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models



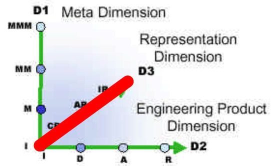
D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

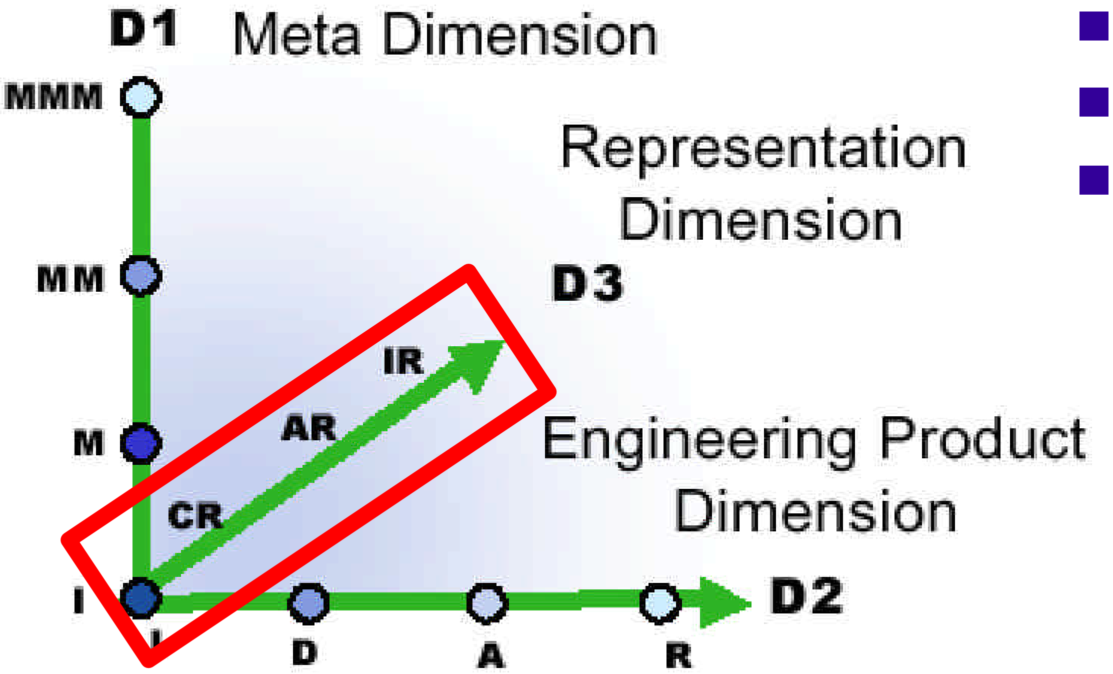

D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
tom : client
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
Feature
Host
Executable
Connectors
Subsystem
Components
Package
Class
Association
AssociationEnd
Boxes
Lines
Symbol
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models


D3
Appliware execution
Software
Appliware
Metaware
D2
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
withdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req
Requirement
Non Funct. Req
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24800
a2204 : Account
balance= 300
Client
name : string
1
*
client
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
• • •
JavaClass
JavaField
JavaMethod
Statement
Expression
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models


D2
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
client
widthdraw
transferMoney
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Non Funct. Req.
Requirement
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
D1
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

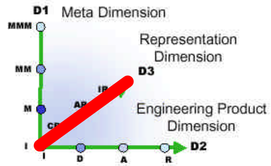
D3: The Representation Dimension





- The Representation Towers
- The Representation Pyramid
- The Representation actors

D3: The Representation Towers



Conceptual Model
↑
Specification Model
↑
Implementation Model

UML – Fowler

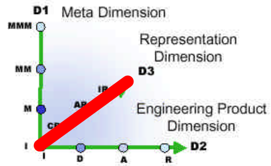
Language
↑
Abstract grammar
↑
Concrete grammar

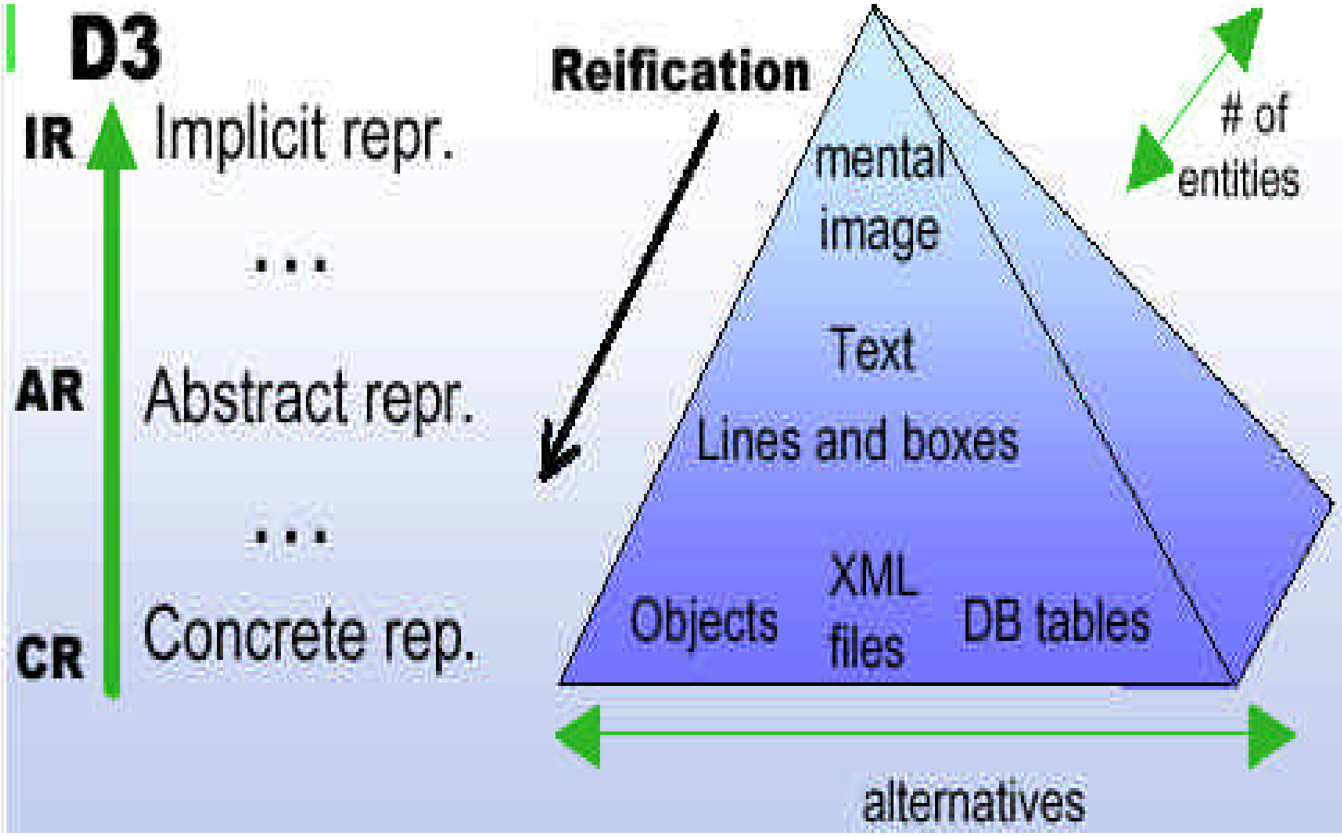
Languages

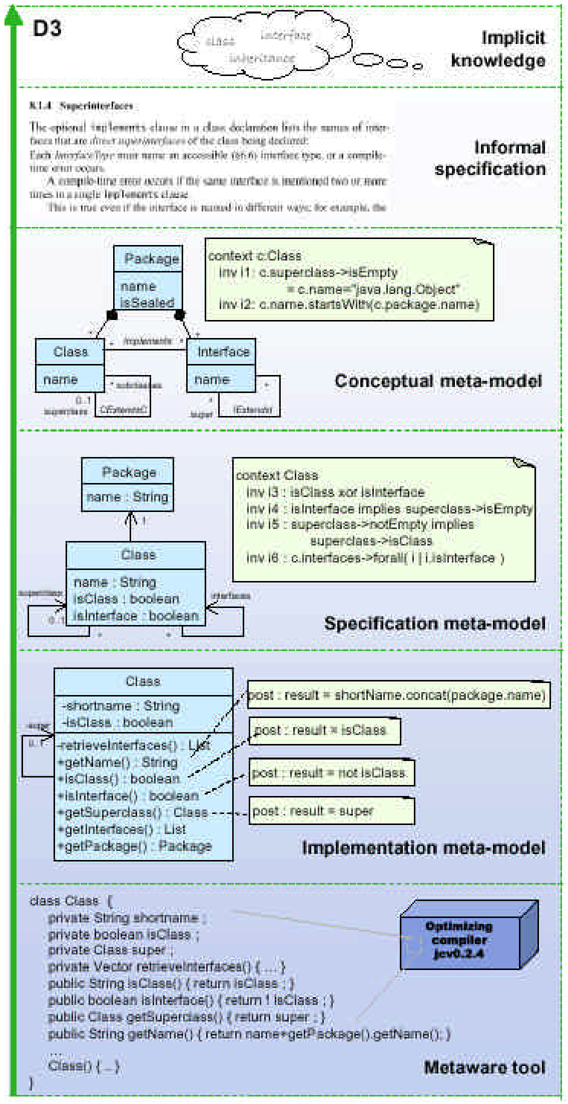
Conceptual Schema
↑
Logical Schema
↑
Physical Schema

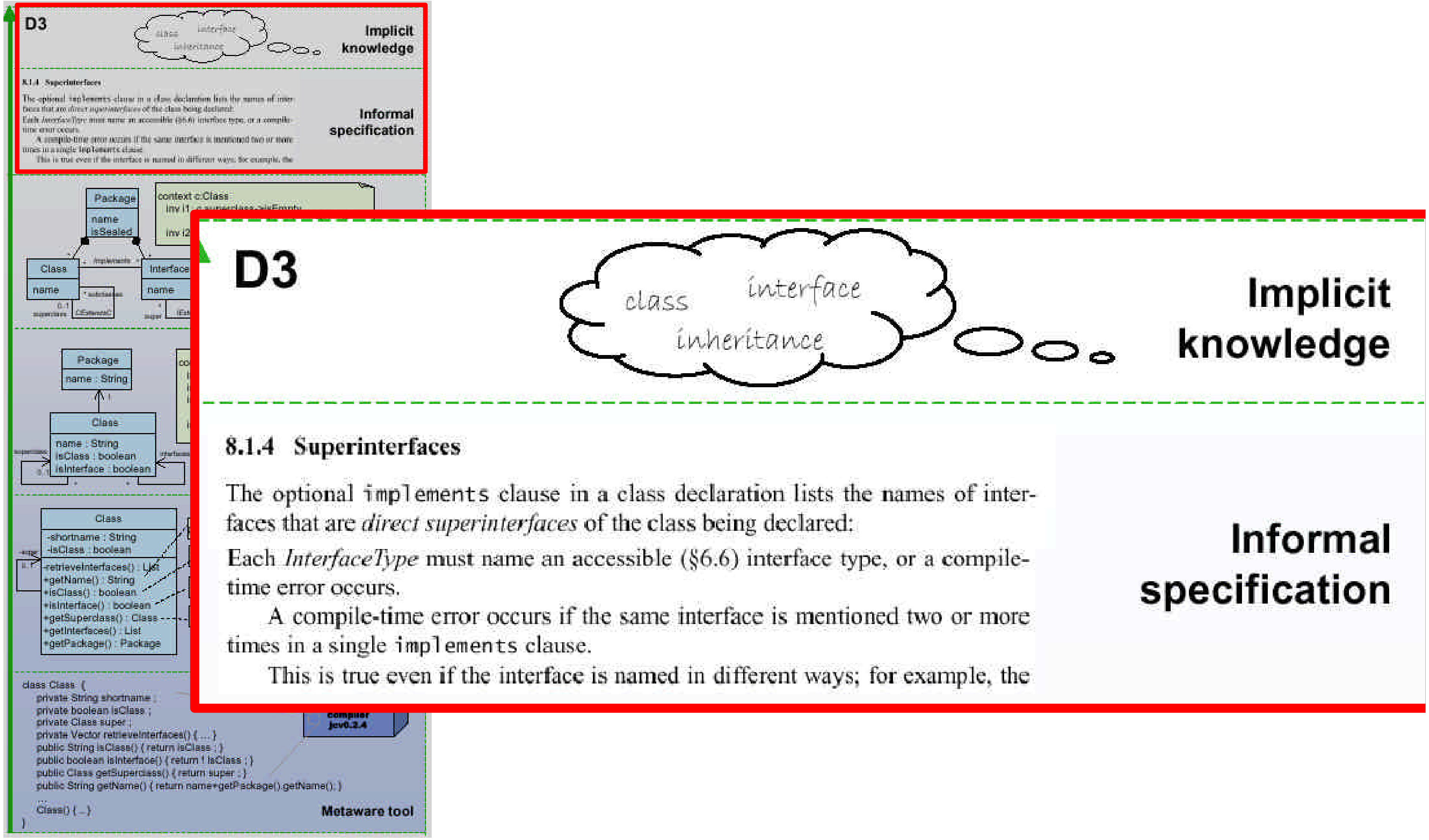
Databases

D3: The Representation Dimension






D3
class
interface
inheritance
Implicit knowledge
Informal specification
Package
name
isSealed
context c:Class
inv i1: c.superclass->isEmpty
= c.name="java.lang.Object"
inv i2: c.name.startsWith(c.package.name)
Class
name
Interface
name
Conceptual meta-model
Package
name : String
context Class
inv i3 : isClass xor isInterface
inv i4 : isInterface implies superclass->isEmpty
inv i5 : superclass->notEmpty implies
superclass->isClass
inv i6 : c.interfaces->forall(i | i.isInterface)
Class
name : String
isClass : boolean
isInterface : boolean
Specification meta-model
Class
-shortname : String
-isClass : boolean
-retrieveInterfaces() : List
+getName() : String
+isClass() : boolean
+isInterface() : boolean
+getSuperclass() : Class
+getInterfaces() : List
+getPackage() : Package
post : result = shortName.concat(package.name)
post : result = isClass
post : result = not isClass
post : result = super
Implementation meta-model
class Class {
private String shortname ;
private boolean isClass ;
private Class super ;
private Vector retrieveInterfaces() { ... }
public String isClass() { return isClass ; }
public boolean isInterface() { return ! isClass ; }
public Class getSuperclass() { return super ; }
public String getName() { return name+getPackage().getName(); }
...
Class() { ... }
}
Optimizing compiler jcv0.2.4
Metaware tool



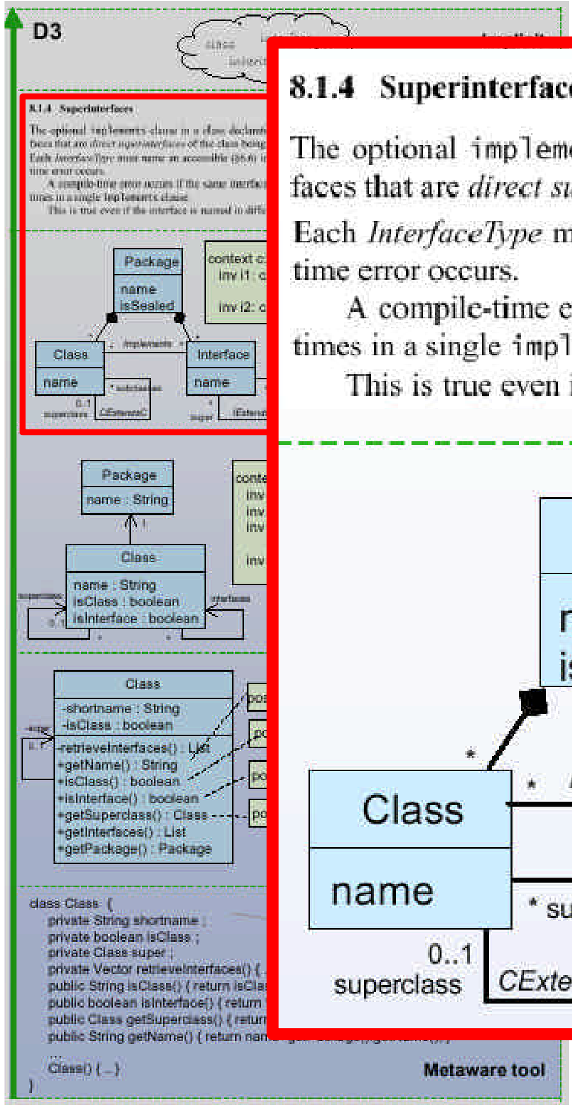
8.1.4 Superinterfaces

The optional `implements` clause in a class declaration lists the names of interfaces that are *direct superinterfaces* of the class being declared:

Each *InterfaceType* must name an accessible (§6.6) interface type, or a compile-time error occurs.

A compile-time error occurs if the same interface is mentioned two or more times in a single `implements` clause.

This is true even if the interface is named in different ways; for example, the



8.1.4 Superinterfaces

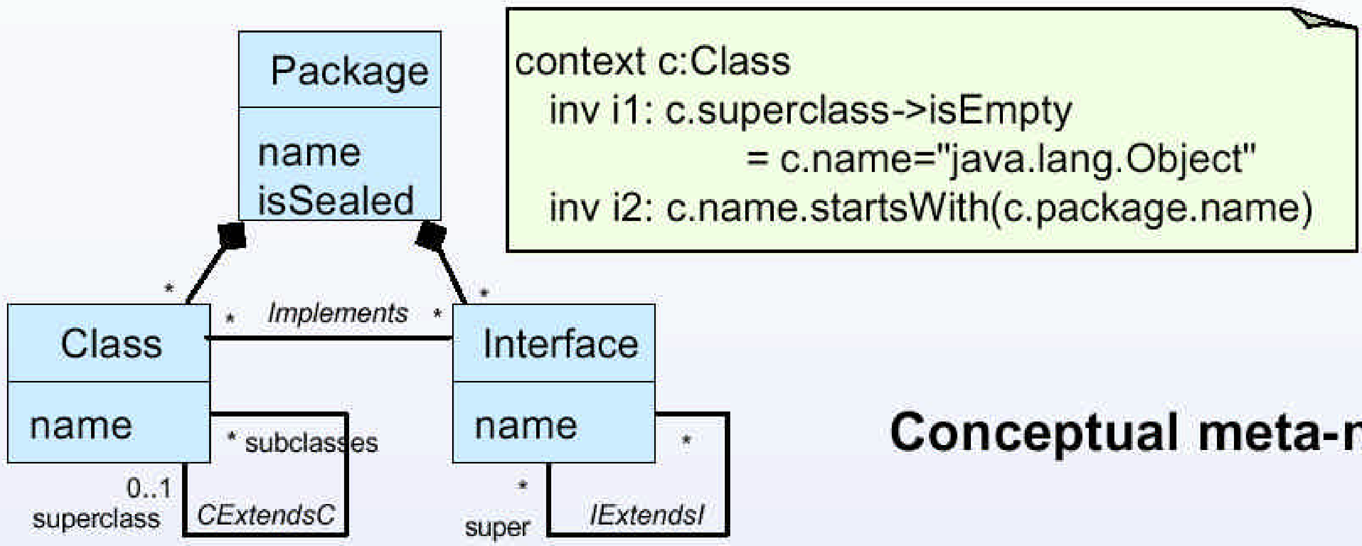
The optional `implements` clause in a class declaration lists the names of interfaces that are *direct superinterfaces* of the class being declared:

Each *InterfaceType* must name an accessible (§6.6) interface type, or a compile-time error occurs.

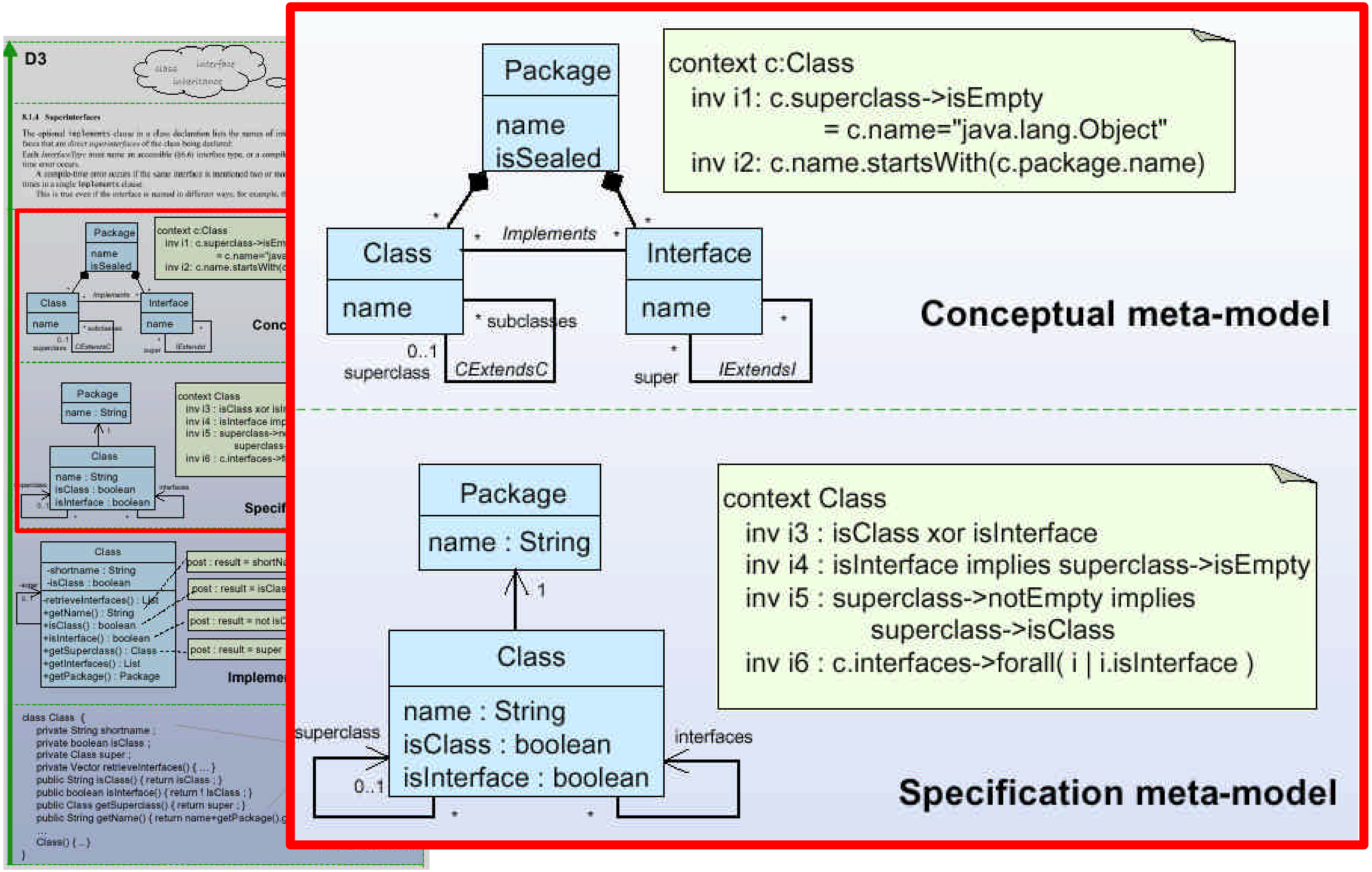
A compile-time error occurs if the same interface is mentioned two or more times in a single `implements` clause.

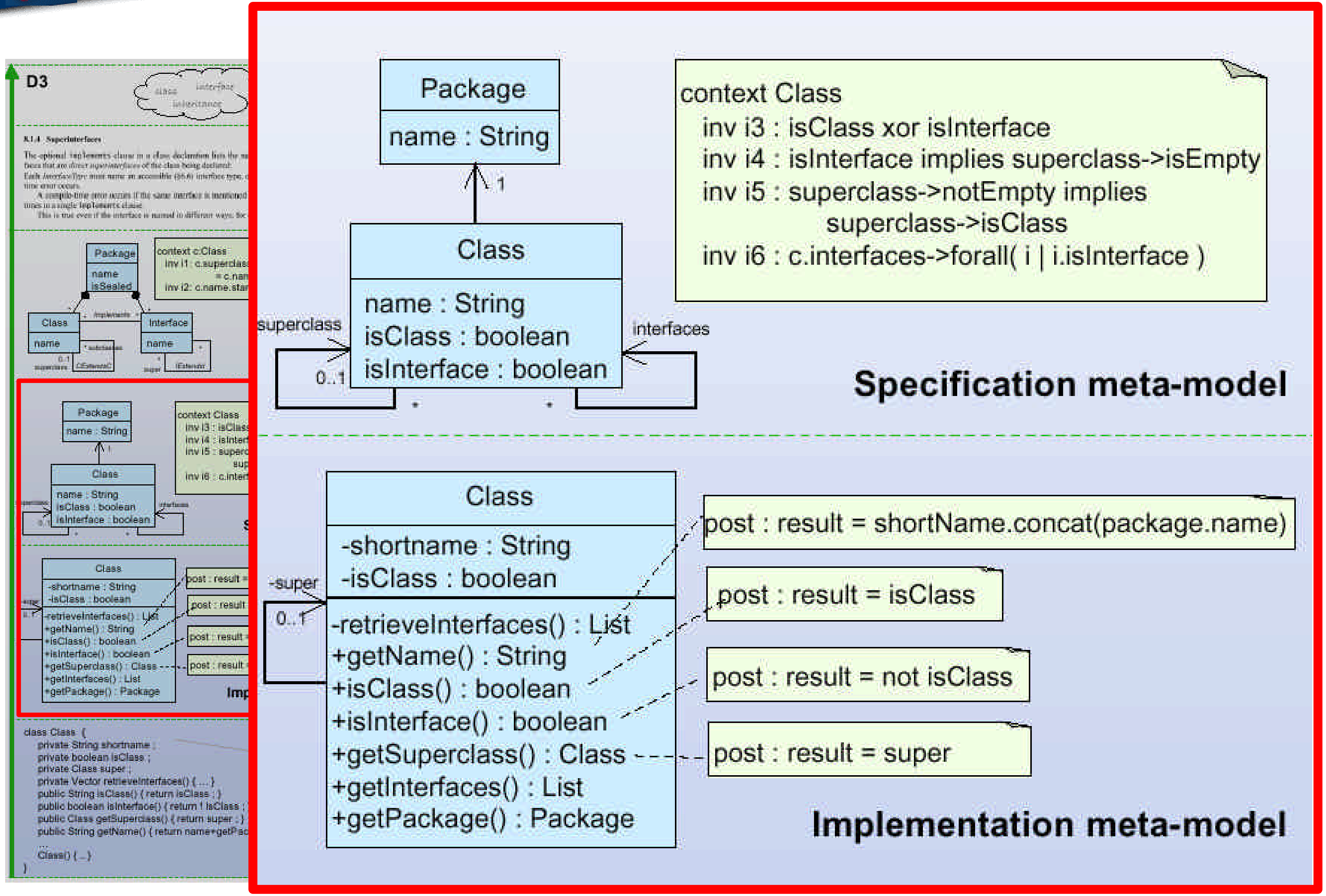
This is true even if the interface is named in different ways; for example, the

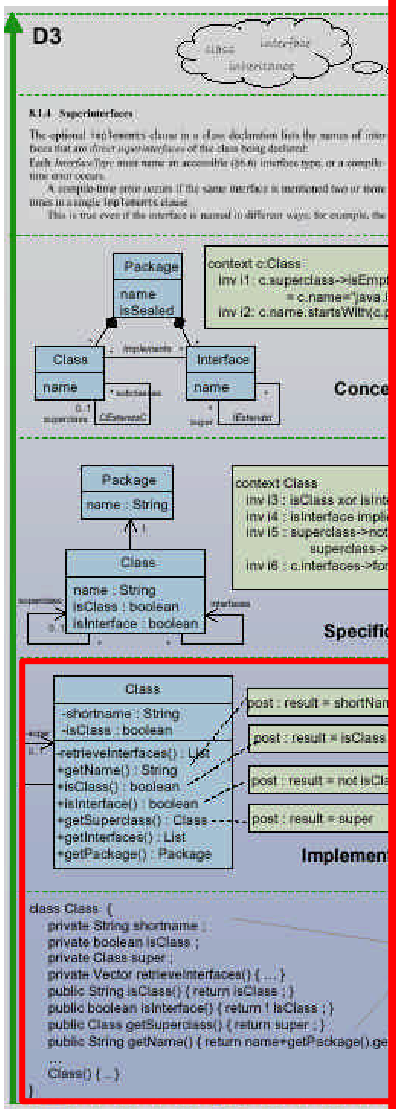
Informal specification

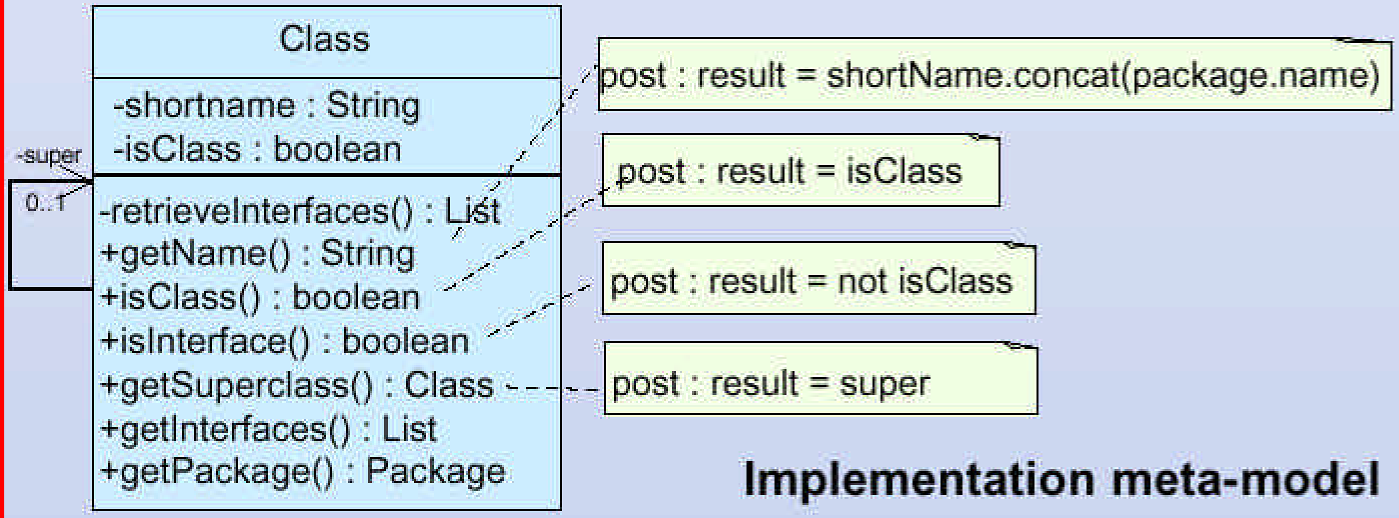


Conceptual meta-model


D3
Package
name
isSealed
context c:Class
inv i1: c.superclass->isEmpty
= c.name="java.lang.Object"
inv i2: c.name.startsWith(c.package.name)
Class
name
Implements
Interface
name
subclasses
0..1
superclass
CExtendsC
super
IExtendsI
Conceptual meta-model
Package
name : String
1
Class
name : String
isClass : boolean
isInterface : boolean
superclass
0..1
interfaces
context Class
inv i3 : isClass xor isInterface
inv i4 : isInterface implies superclass->isEmpty
inv i5 : superclass->notEmpty implies
superclass->isClass
inv i6 : c.interfaces->forall(i | i.isInterface)
Specification meta-model

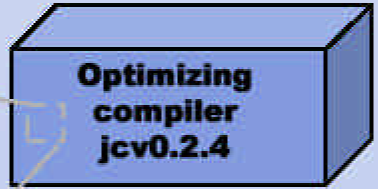

Package
name : String
1
context Class
inv i3 : isClass xor isInterface
inv i4 : isInterface implies superclass->isEmpty
inv i5 : superclass->notEmpty implies
superclass->isClass
inv i6 : c.interfaces->forall(i | i.isInterface)
Class
name : String
isClass : boolean
isInterface : boolean
superclass
0..1
interfaces
*
*
Specification meta-model
Class
-shortname : String
-isClass : boolean
-retrieveInterfaces() : List
+getName() : String
+isClass() : boolean
+isInterface() : boolean
+getSuperclass() : Class
+getInterfaces() : List
+getPackage() : Package
-super
0..1
post : result = shortName.concat(package.name)
post : result = isClass
post : result = not isClass
post : result = super
Implementation meta-model



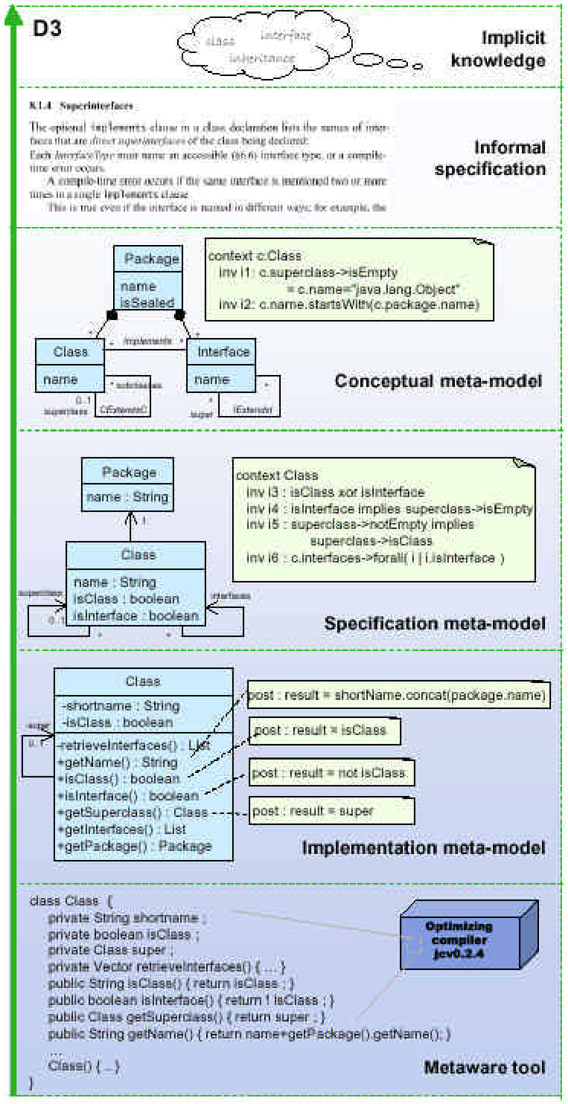


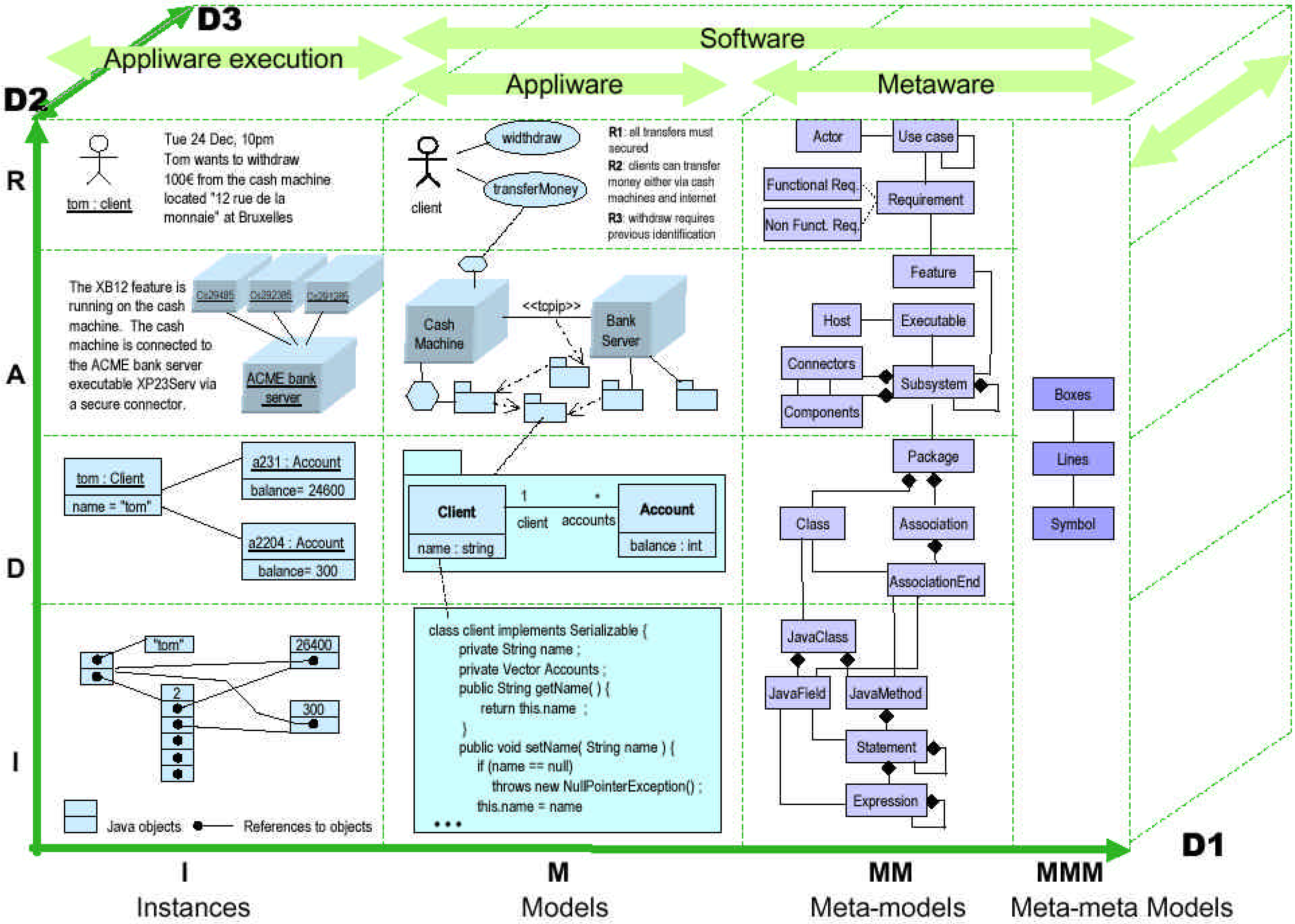
Implementation meta-model

```
class Class {
    private String shortname ;
    private boolean isClass ;
    private Class super ;
    private Vector retrieveInterfaces() { … }
    public String isClass() { return isClass ; }
    public boolean isInterface() { return ! IsClass ; }
    public Class getSuperclass() { return super ; }
    public String getName() { return name+getPackage().getName(); }
    …
    Class() { .. }
}
```



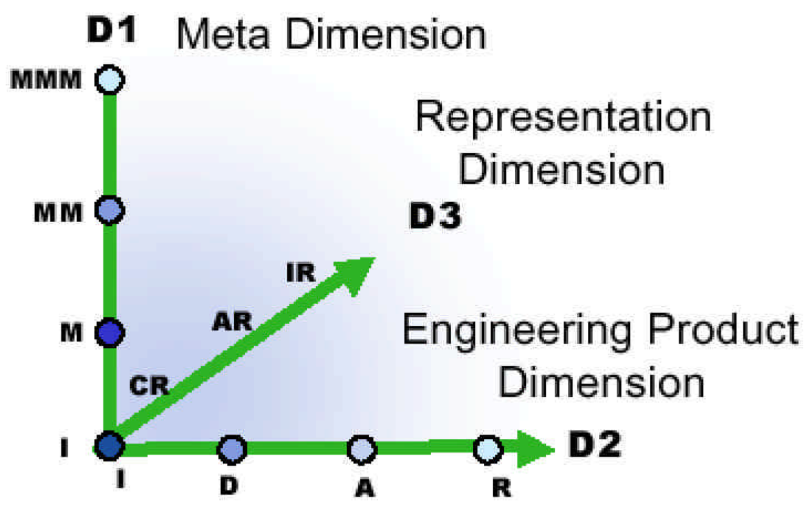
Metaware tool


D3
Implicit knowledge
Informal specification
Package
name
isSealed
Class
name
Interface
name
context c:Class
inv i1: c.superclass->isEmpty
= c.name="java.lang.Object"
inv i2: c.name.startsWith(c.package.name)
Conceptual meta-model
Package
name : String
Class
name : String
isClass : boolean
isInterface : boolean
context Class
inv i3 : isClass xor isInterface
inv i4 : isInterface implies superclass->isEmpty
inv i5 : superclass->notEmpty implies
superclass->isClass
inv i6 : c.interfaces->forall(i | i.isInterface)
Specification meta-model
Class
-shortname : String
-isClass : boolean
-retrieveInterfaces() : List
+getName() : String
+isClass() : boolean
+isInterface() : boolean
+getSuperclass() : Class
+getInterfaces() : List
+getPackage() : Package
post : result = shortName.concat(package.name)
post : result = isClass
post : result = not isClass
post : result = super
Implementation meta-model
class Class {
private String shortname ;
private boolean isClass ;
private Class super ;
private Vector retrieveInterfaces() { ... }
public String isClass() { return isClass ; }
public boolean isInterface() { return ! isClass ; }
public Class getSuperclass() { return super ; }
public String getName() { return name+getPackage().getName(); }
...
Class() { ... }
}
Optimizing compiler jcv0.2.4
Metaware tool


D3
D2
D1
Appliware execution
Software
Appliware
Metaware
R
A
D
I
tom : client
Tue 24 Dec, 10pm
Tom wants to withdraw
100€ from the cash machine
located "12 rue de la
monnaie" at Bruxelles
widthdraw
transferMoney
client
R1: all transfers must secured
R2: clients can transfer money either via cash machines and internet
R3: withdraw requires previous identification
Actor
Use case
Functional Req.
Requirement
Non Funct. Req.
The XB12 feature is running on the cash machine. The cash machine is connected to the ACME bank server executable XP23Serv via a secure connector.
ACME bank server
Cash Machine
<<tcpip>>
Bank Server
Feature
Host
Executable
Connectors
Subsystem
Components
Boxes
Lines
Symbol
tom : Client
name = "tom"
a231 : Account
balance= 24600
a2204 : Account
balance= 300
Client
name : string
1
client
*
accounts
Account
balance : int
Package
Class
Association
AssociationEnd
"tom"
26400
2
300
Java objects
References to objects
class client implements Serializable {
private String name ;
private Vector Accounts ;
public String getName( ) {
return this.name ;
}
public void setName(String name) {
if (name == null)
throws new NullPointerException() ;
this.name = name
...
JavaClass
JavaField
JavaMethod
Statement
Expression
I
Instances
M
Models
MM
Meta-models
MMM
Meta-meta Models

Classifying Software Artefacts

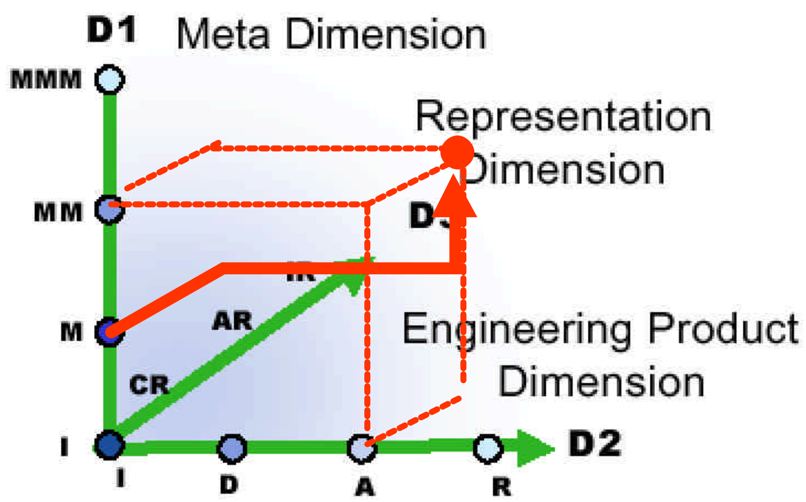
- Using the 3D Framework to classify software artefacts
- Coordinates in the reverse order D3-D2-D1



- Examples :
  - CR-A-MM
  - AR-D-M
  - …

Classifying Software Transformations

- Transformations or processes as paths in 3D
- Useful to classify SE tools and methods

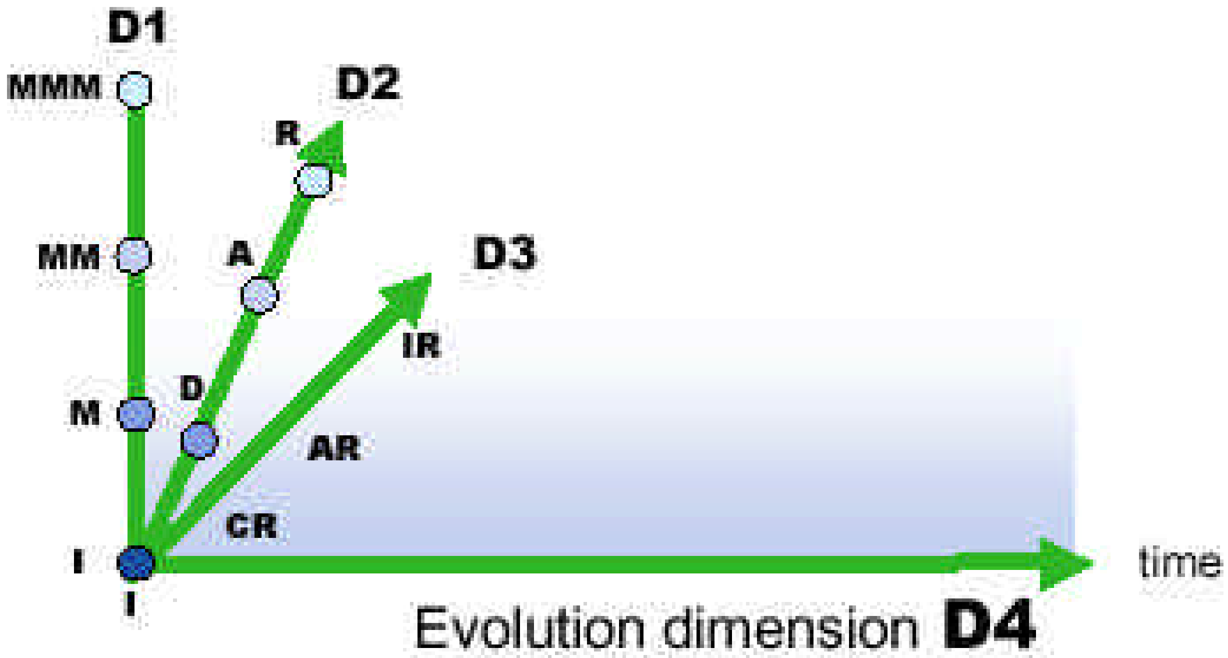


- Forward engineering
- Reverse engineering
- Evolution & co-evolution

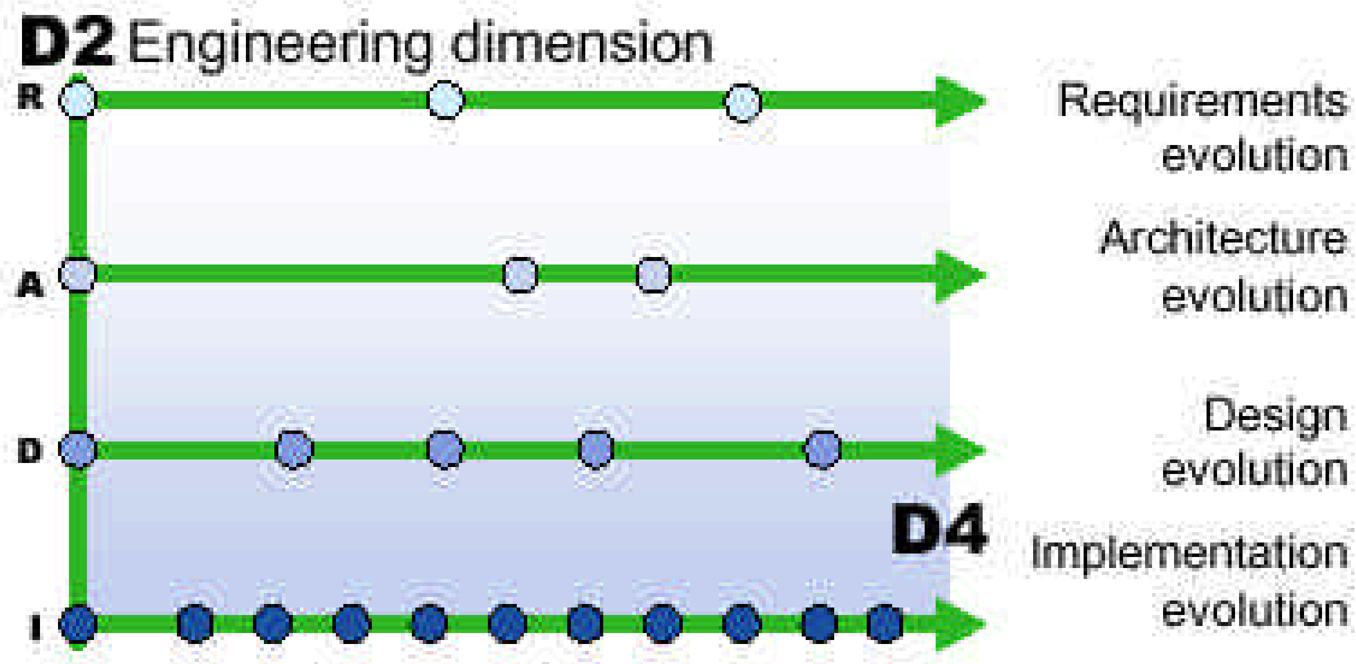
Evolution :

Entering the fourth dimension…

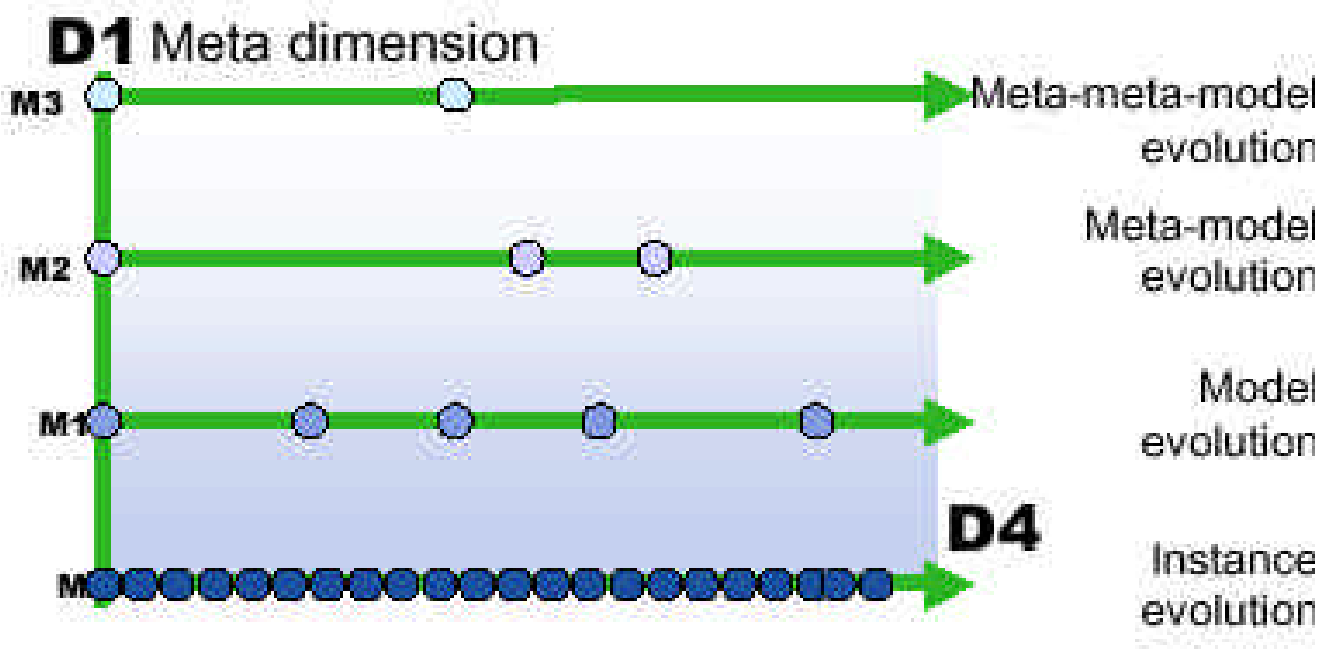
Evolution: Entering the Fourth Dimension (D4)



Co-evolution along the engineering dimension D2+D4



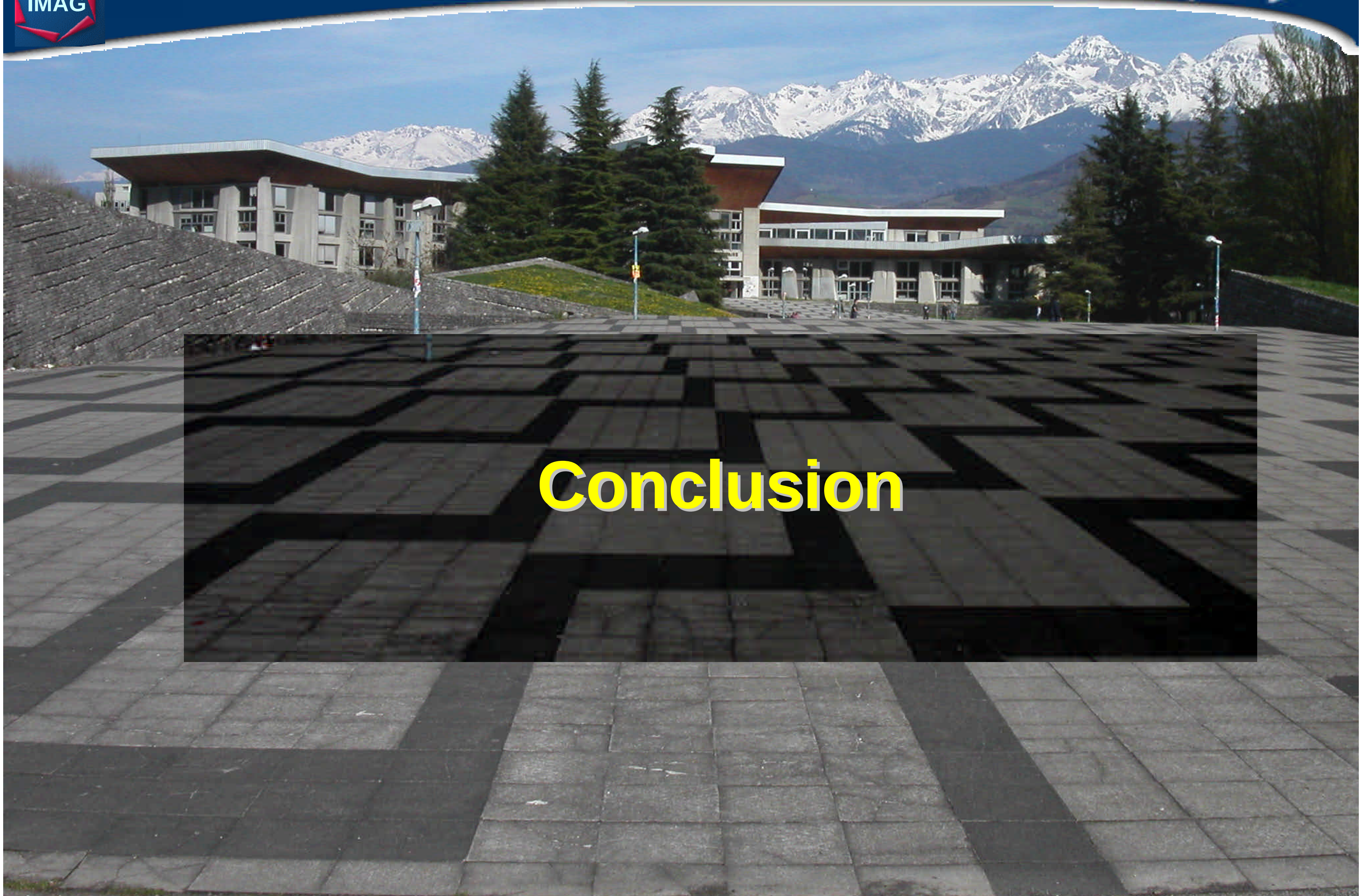
Co-evolution along the meta dimension D1+D4



Meta-model / model co-evolution at DS

- Incremental definition of a proprietary component technology
- Incremental implementation of tools by the tool support team
- Production of component-based software at the same time

- Meta-models should be versionned
- Different variants of the meta-model used
  - by different teams within DS
  - by partner companies
- Co-evolution managed in ad-hoc way
- Manual or semi-automatic transformation



Conclusion

Conclusion

- Many academic issues related to meta-modeling / meta-programming
- More issues coming from industry
- Co-evolution of meta-models and models
- Reverse engineering meta-models
- Tool support is required

Supporting evolution at various level
is an important requirement
for the success of
model-driven approaches (e.g. MDA)

Meta-model for evolution vs. evolution of meta-models

Call For Papers

First International Workshop on Meta-models and Schemas for Reverse Engineering

November 13, 2003, Victoria, BC, Canada

www-adele.imag.fr/atem2003

With WCRE'2003

Organizers
Jean-Marie Favre, University of Grenoble, France
Mike Godfrey, University of Waterloo, Canada
Andreas Winter, University Koblenz-Landau, Germany