



A Feature-based Comparison of Melanee and MetaDepth

Ralph Gerbig¹, Colin Atkinson¹, Juan de Lara², Esther Guerra²

¹University of Mannheim

²Universidad Autónoma de Madrid

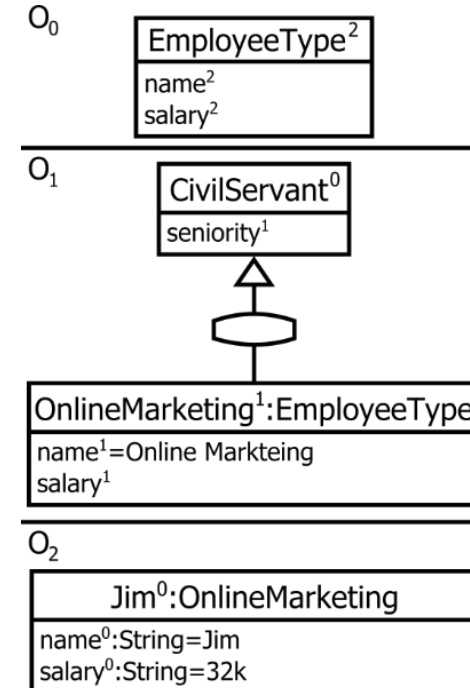
Obvious difference?



- Text based (HUTN-dialect) vs. diagrammatic (UML/ER dialect)
- Programming vs. language engineering
- Standalone console application vs. Eclipse distribution
- Not EMF-based vs EMF-based

```
1 Model O0@2{
2   Node EmployeeType{
3     name:String;
4     salary:String;
5   }
6 }
7
8 O0 O1{
9   abstract EmployeeType CivilServant{
10    seniority:String
11  }
12  EmployeeType OnlineMarketing: CivilServant{
13    name="Online Marketing";
14  }
15 }
16
17 O1 O2{
18   OnlineMarketing Jim{
19     name="Jim";
20     salary="32k";
21   }
22 }
```

MetaDepth



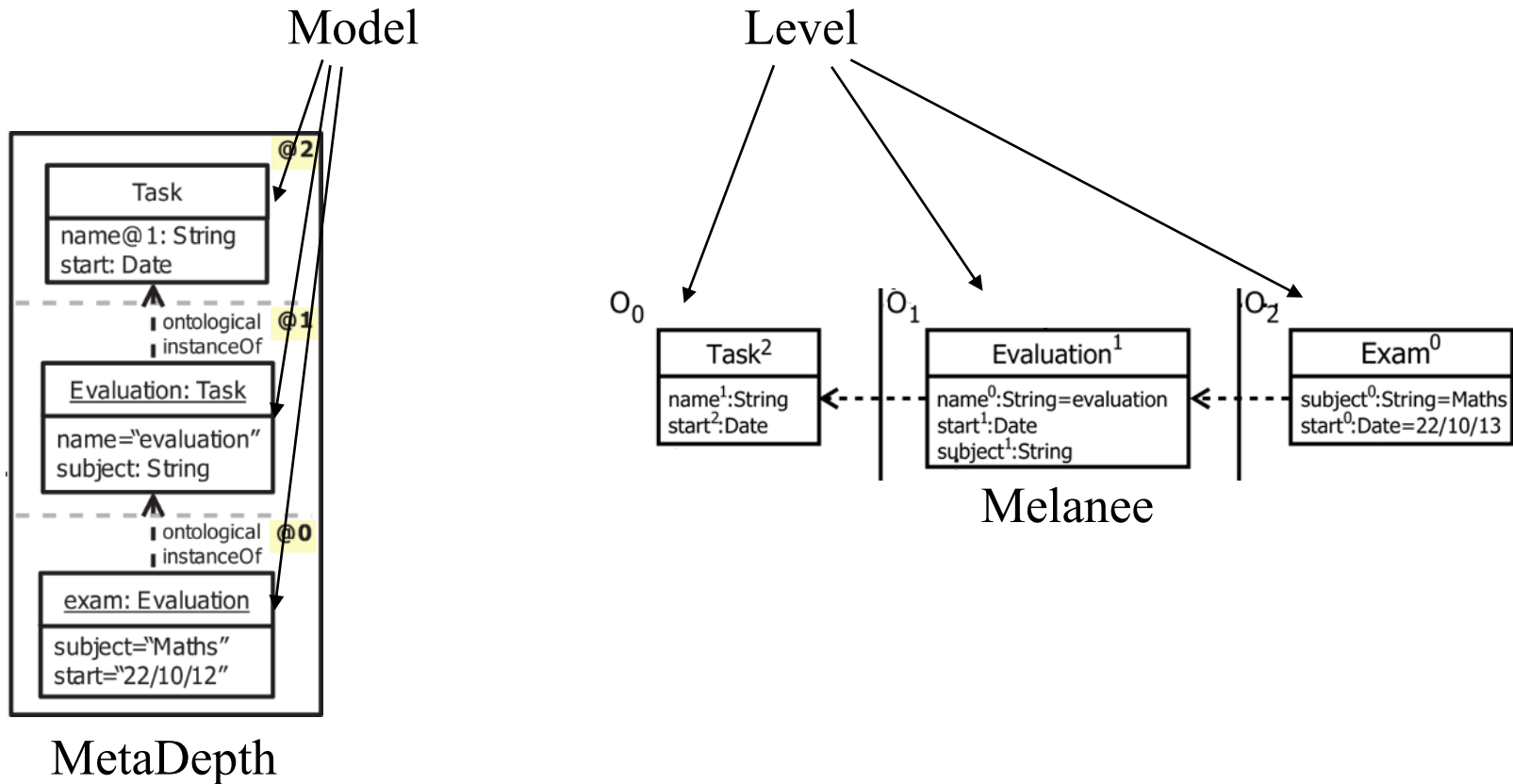
Melanee



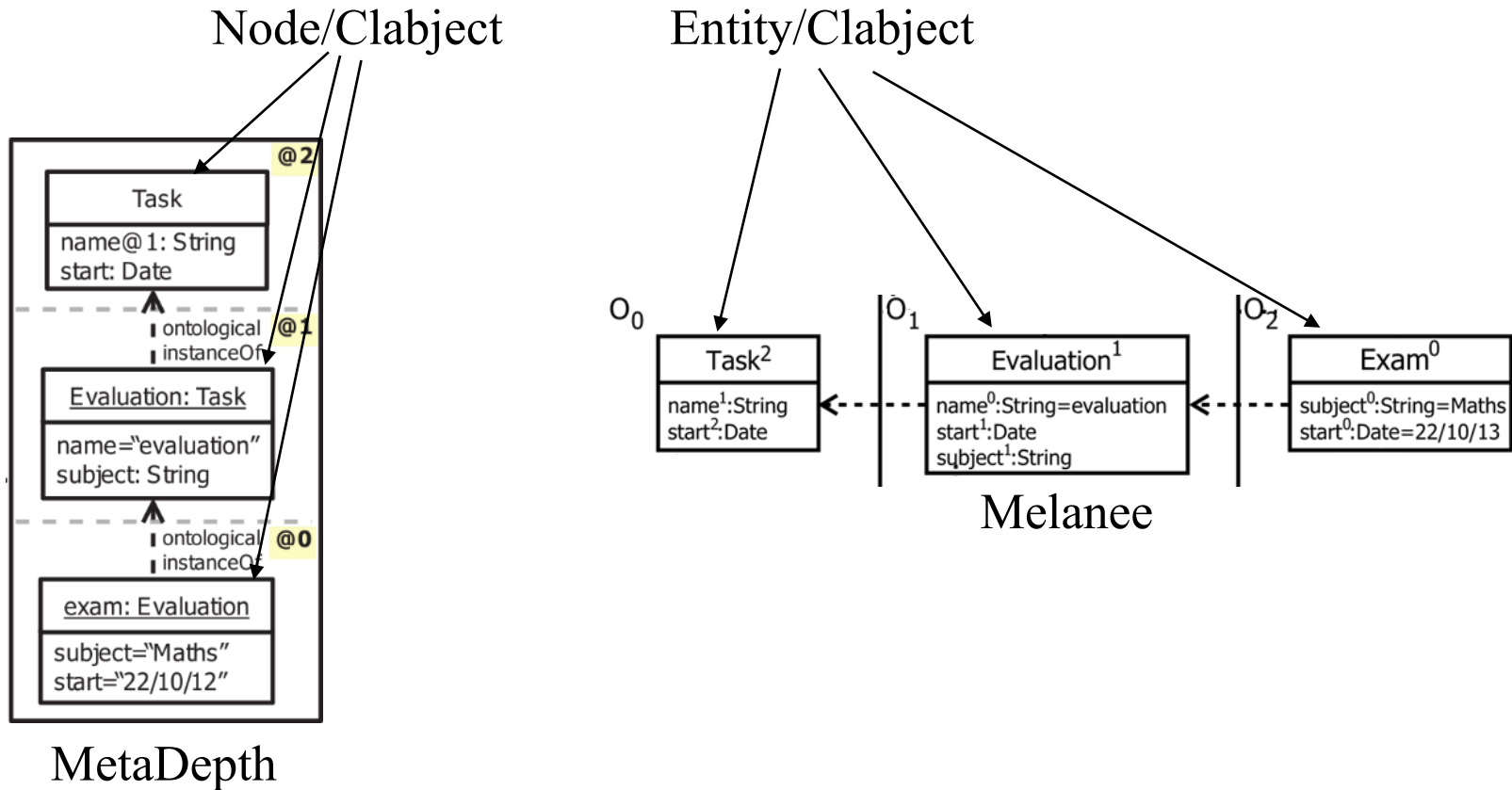
- Equal terminology is highlighted
 - Only the term potency and clabject are equally used
- Deep Model and Mutability are not available in MetaDepth

MetaDepth	Melanee
-	Deep Model
Model	Level
Clabject	Clabject
Node	Entity
Edge	Connection
Reference	-
Field	Attribute
Potency	Potency
Field Potency	Durability
-	Mutability

- Notational difference between @-notation and superscript notation for expressing potency

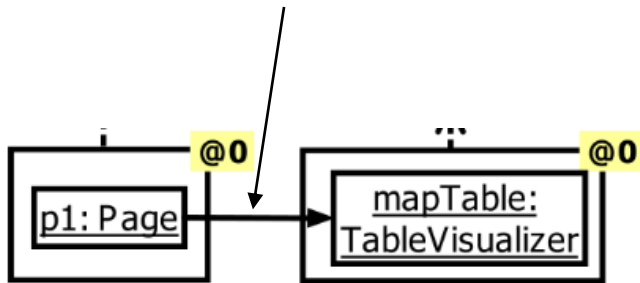


- Notational difference between @-notation and superscript notation for expressing potency



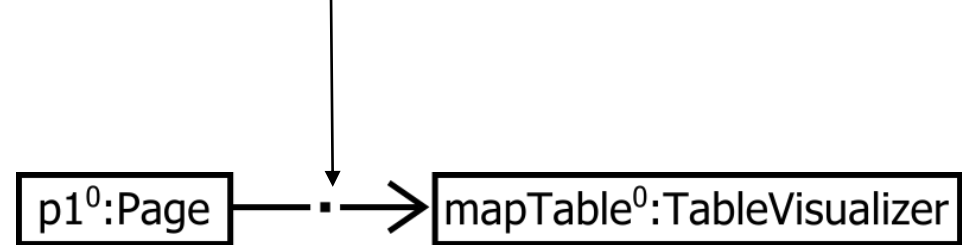
- Notational difference between @-notation and superscript notation for expressing potency

Edge/Reference/Clabject



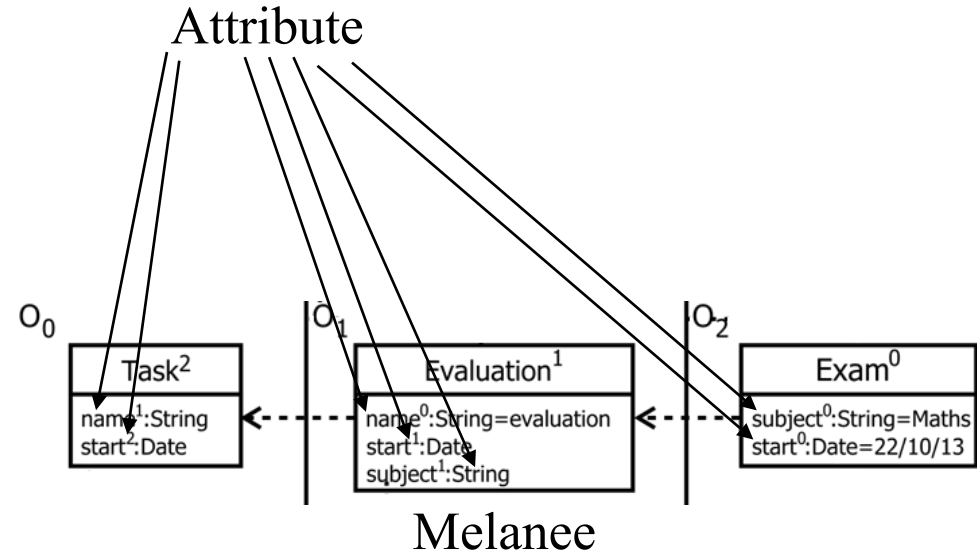
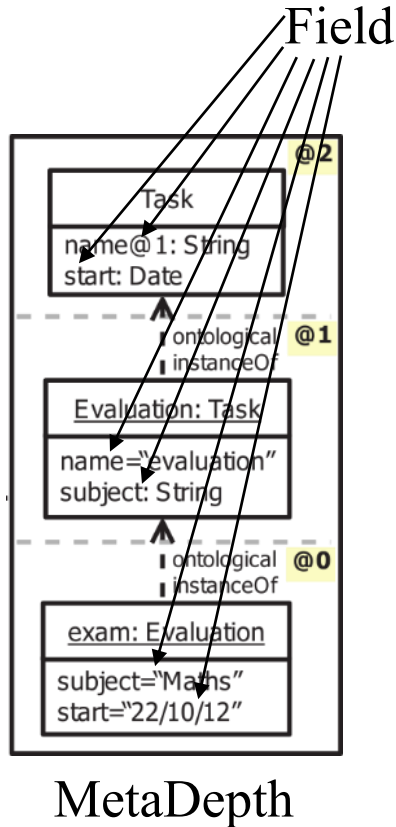
MetaDepth

Connection/Clabject

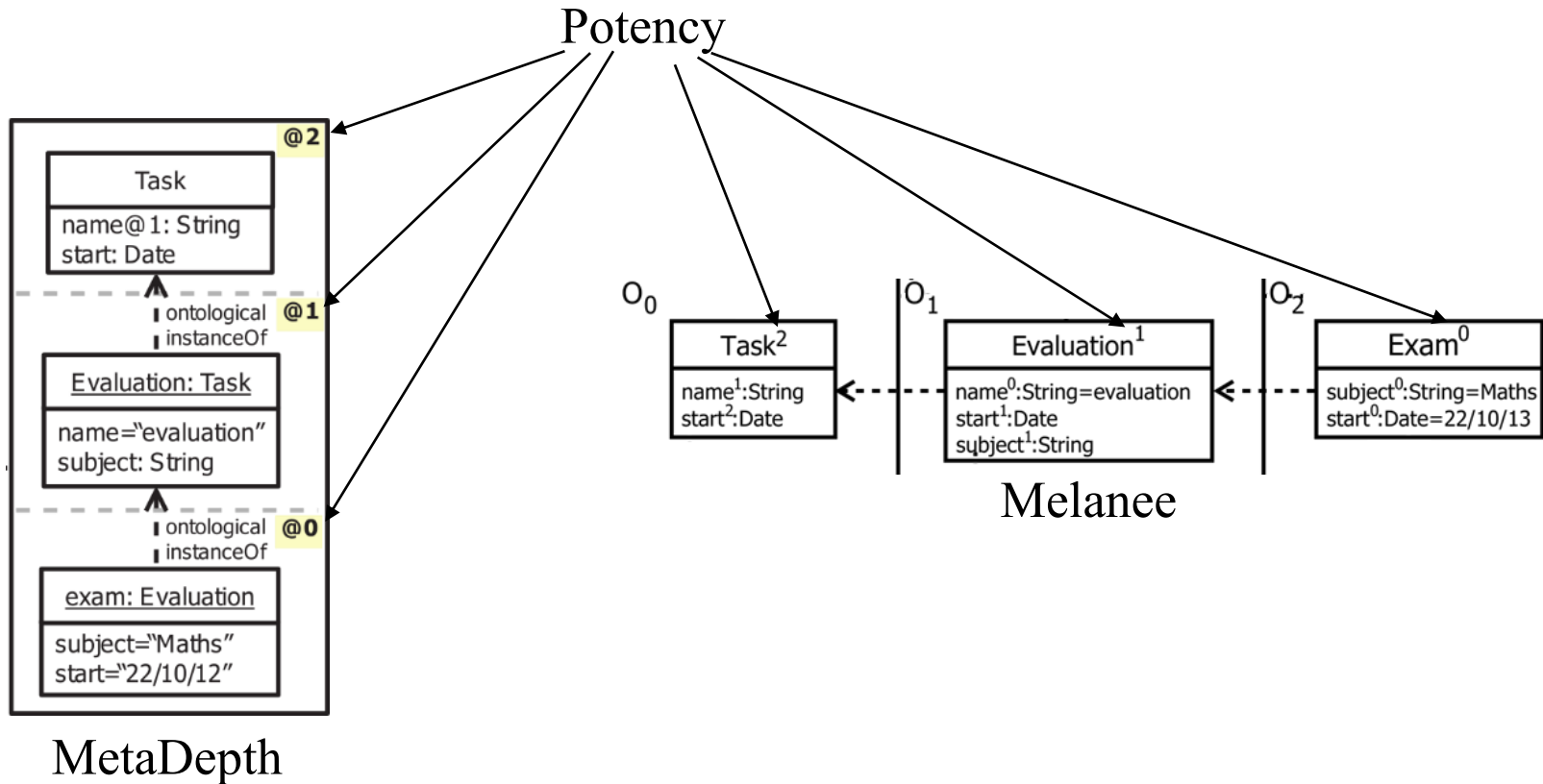


Melancee

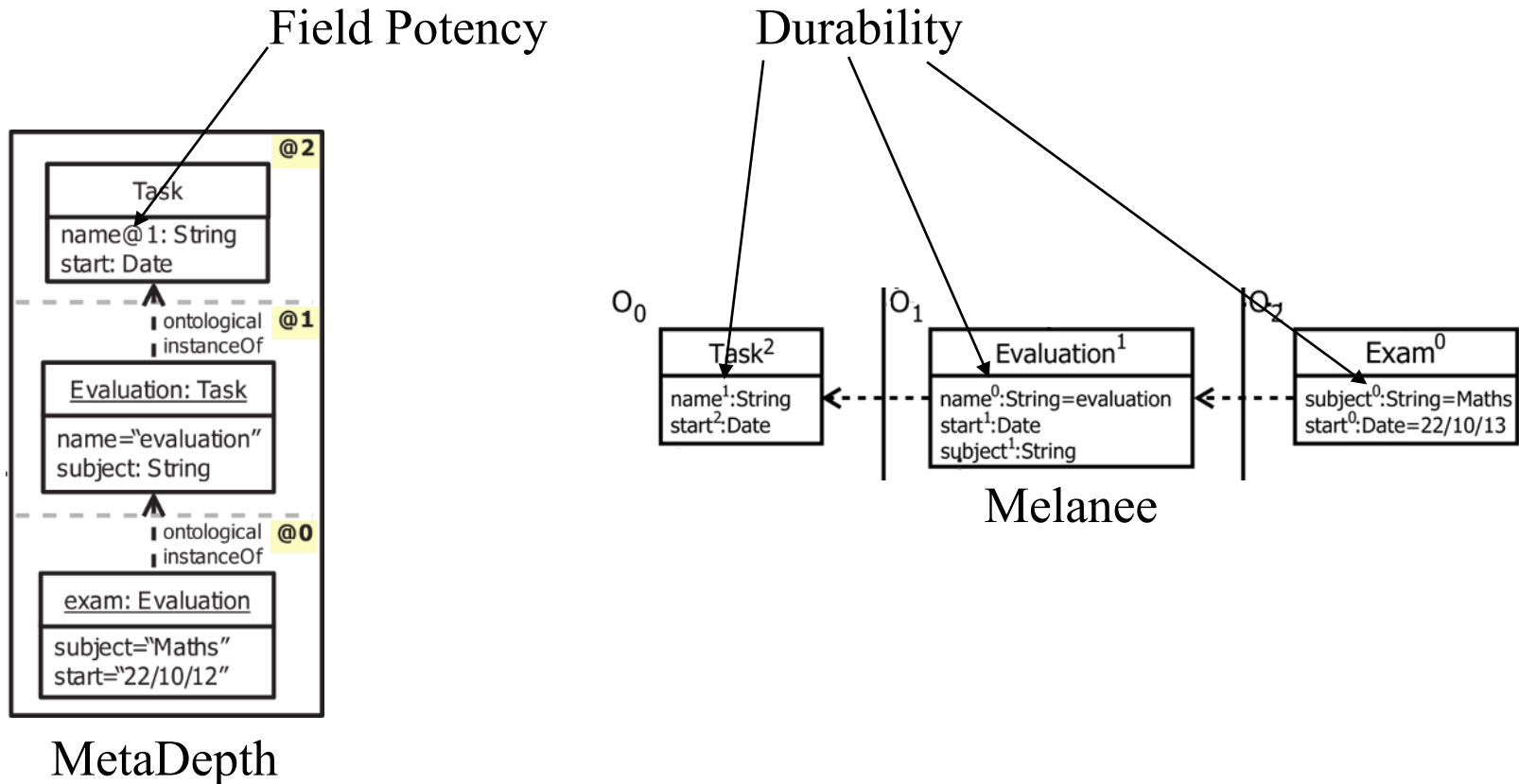
- Notational difference between @-notation and superscript notation for expressing potency



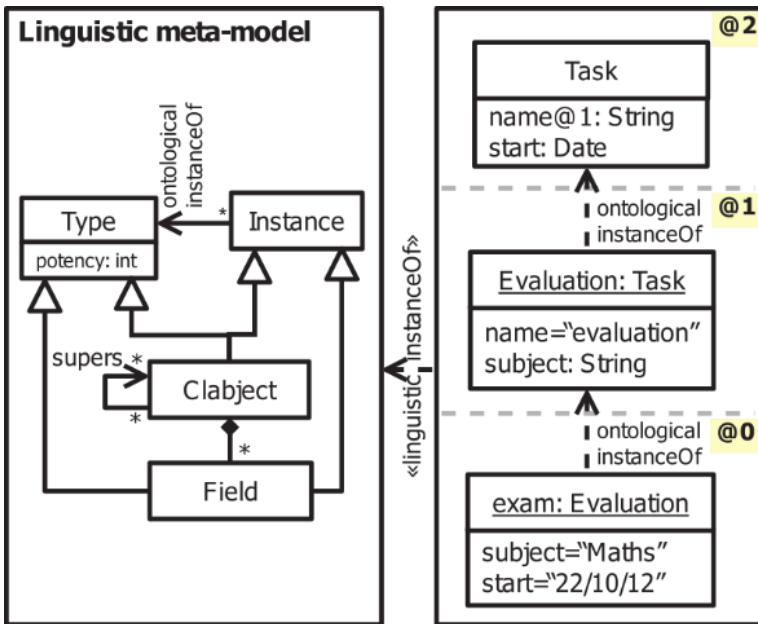
- Notational difference between @-notation and superscript notation for expressing potency



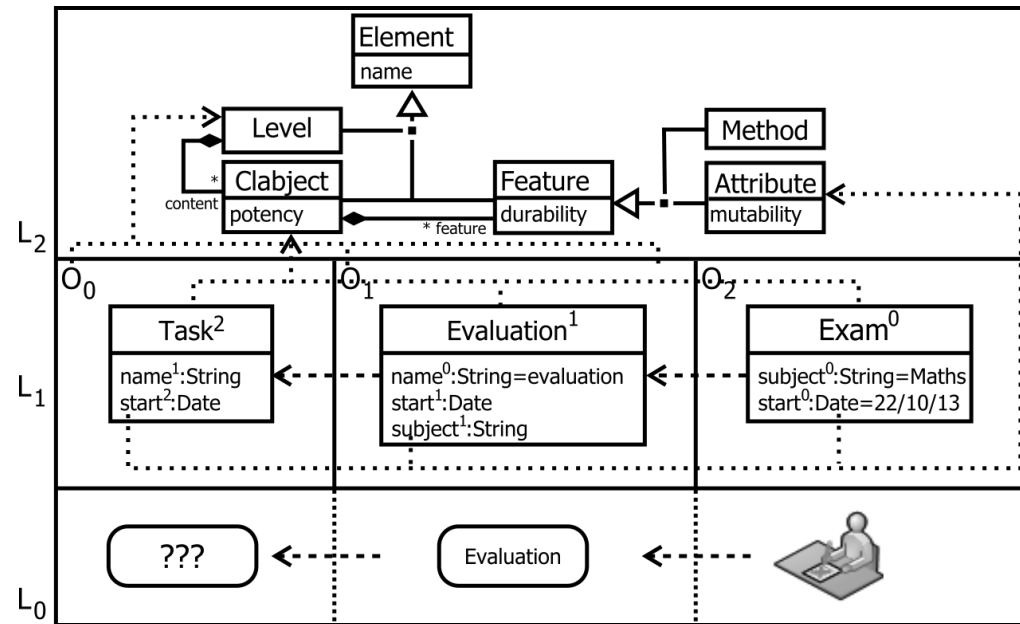
- Notational difference between @-notation and superscript notation for expressing field potency / durability



- Orthogonal Classification Architecture
 - Linguistic and Ontological Classification
- Potency for Levels (MetaDepth only), Clajects and Attributes
 - Potency is reduced at instances by one
 - Support of *-potency
 - Potency 0 clajects cannot have instances
 - No mutability in MetaDepth

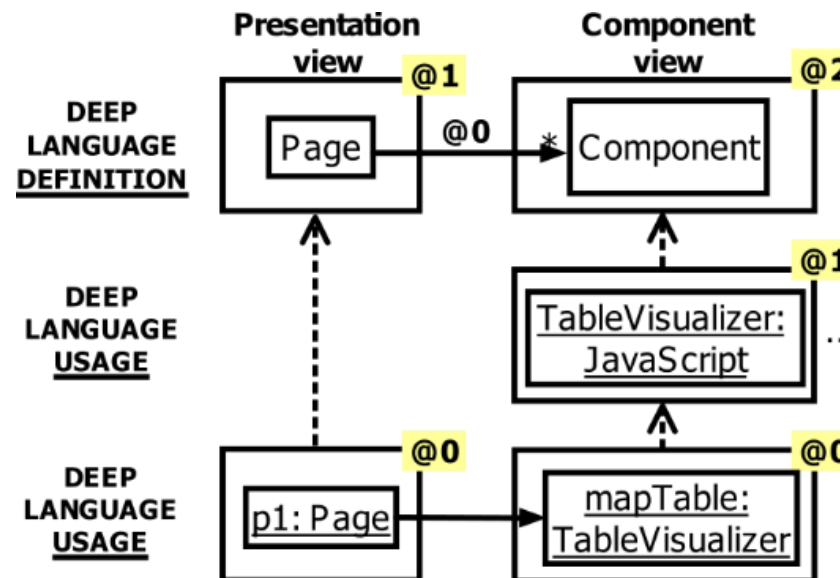


MetaDepth

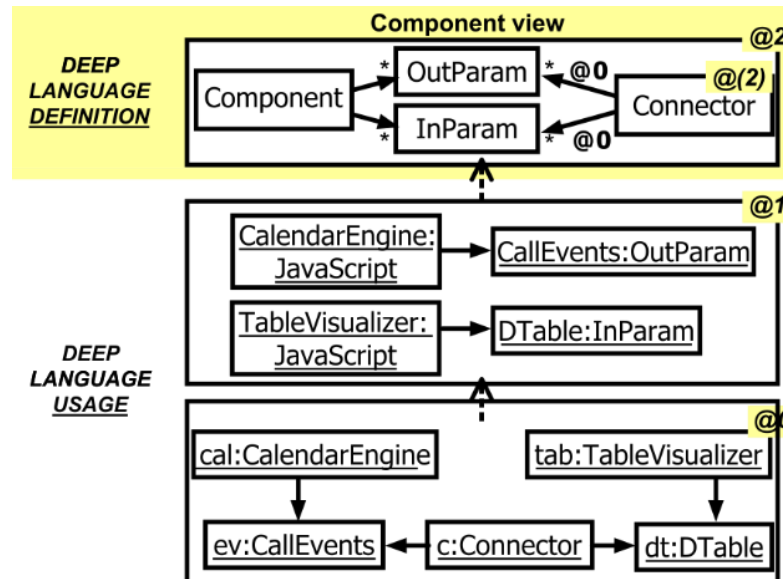


Melanee

- Supported in MetaDepth only
- Of advantage in model import scenarios
 - Page and Component are defined in different models
- Clobjects of different potency can be connected with each other
 - Page has potency of 1 and Component of 2
- It can be specified that Page references indirect instances of Component with potency 0
 - @0 annotation at connection

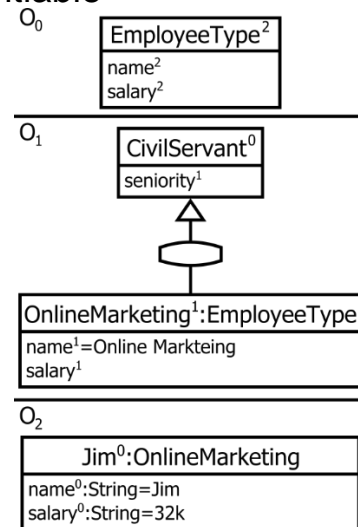


- Supported in MetaDepth only
- Defines at which classification level an instance of a clabject has to exist
- Prevents identity instantiation
 - Instantiation just for the reason of being able to create instances on classification level further down
- Connector is from @2-level is used at @0-level without instantiation on @1-level
 - Leap potency of 2 in combination of deep references to parameters



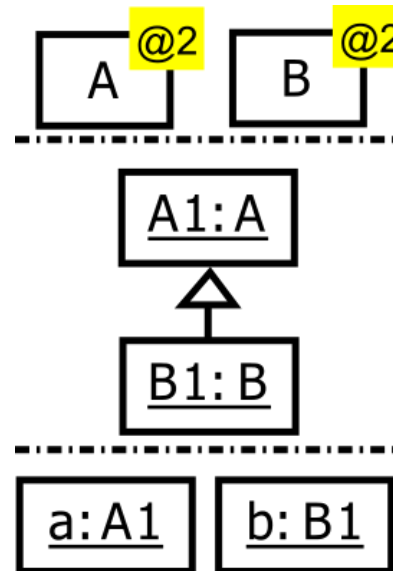


- Example does not work in MetaDepth but in Melanee
- In MetaDepth subtype and supertypes need to have “compatible” types
 - → CivilServant has to have EmployeeType as type
- In MetaDepth potency of all clabjects must be one lower than type
 - → CivilServant must have potency 1 and be defined as abstract
- Abstract vs. Potency 0 in MetaDepth
 - Potency 0 → a clabject is an instance only and can thus not be instantiated. It cannot have subclasses
 - Abstract → a clabject with potency higher one that cannot have direct instances. Only concrete subclasses are instantiable





- Example does not work in MetaDepth but in Melanee
- A1 and B1 do have disjoint ontological types
- **Problem:** A.allInstances() evaluated at level 0 would return b (an “indirect” instance of B), which would be unexpected
 - → In MetaDepth subtype and supertypes need to have “compatible” types
 - → A and B need to inherit from each other



- Highlighted rows are equal in Melanee and MetaDepth

	METADEPTH	Melanee	
Potency	Potency Reduction	<ul style="list-style-type: none"> – Potency at instance one equal to METADEPTH lower than at ontological type level – potency 0 clabjects cannot have instance 	
	Leap Potency	available	not available
	Star Potency	available	available
	Strict Meta Modeling	<ul style="list-style-type: none"> – leap potency – deep references 	enforced
	Potency Declaration	<ul style="list-style-type: none"> – At level, clabject – clabjects inherited potency from level if not stated otherwise 	At each clabject
Abstract Keyword	available	no keyword, potency of 0	
Distributed Modeling	importing of Levels (i.e. Models in METADEPTH)	linking to remote Clabjects and Packages	
Classification	Clabject Conformance	<ul style="list-style-type: none"> – potency – attributes – edges & references 	equal to METADEPTH
	Multiple Ontological Classification	dynamic retyping [11]	available
	Linguistic Extension	allowed	allowed



	METADEPTH	Melanee
Durability	derived from Clabject, but can be set individually	set independent from Clabject
Mutability	not available	available
Repeated at instances	not mandatory	mandatory
Default Values at Intermediate Levels	available	available
Association Classes	available	only available kind of connections
References (complex attributes)	available	not available
Cardinality	effects immediate level below	effects all levels below
Containment & Owning	not available	available
Association Types	plain associations only	containment, aggregation, plain association
Connection Diversification	supported	supported
Enumerations	supported	supported
Operations	not available	available
Distributed Modeling	importing of Levels (i.e. Models in METADEPTH)	linking to remote Clabjects and Packages

Attributes, Edges & Distributed Modeling

	METADEPTH	Melanee
Potency	subtype potency equal or equal to METADEPTH higher supertype	
Inheritance	Disjoint Types of super and subtype	not allowed
	Untyped supertype for typed subtype	not allowed
	Multiple Inheritance	supported
	Generalization Sets	not supported
	Attribute Overriding	by duplication
		equal to METADEPTH

- Most significant differences are related to
 - Model vs Level
 - Distributed Modelling
 - Classification
 - Leap Potency
 - Inheritance
 - Abstract Clabjects
 - Connections
 - Deep References
 - Connections and References
 - Inheritance
 - Generalization Sets



- At first glance Melanee and MetaDepth seem to be quite similar
 - The tools however, have significant differences
- We embedded MetaDepth like syntax into the Melanee Tooling
 - This is, however, not 100% compatible to MetaDepth caused by the differences
 - Working on a Melanee to MetaDepth compatibility mode
- Try out the tools
 - MetaDepth --- <http://www.metadepth.org>
 - Melanee --- <http://www.melanee.org>

Thank You!

