

# Historisch- Kulturwissenschaftliche Informationsverarbeitung Woche 5

## UML Aufgaben UML und OMG



# Modellierung – Aufgaben

## Hintergrund

Ein Kunstmuseum plant die Entwicklung eines Bildverwaltungssystems für Architektur, Monumente und andere Kunstgegenstände.

Durch eine Informationsveranstaltung mit der Museumsleiterin haben Sie eine kurze Einführung in die Problematik erhalten.



# Modellierung – Aufgaben

## Problembeschreibung

Das Museum wünscht ein System, mit Hilfe dessen sowohl Museumsangestellte, als auch interessierte Bürgerinnen und Bürger Bilder von Kunstwerken einreichen können.

Das System soll zum einen dazu dienen, digitale Bilder über eine einfache Schnittstelle einzureichen.

Zum anderen sollen Scan-Stationen für analoge Positiv- und Negativbilder in den Arbeitsprozess einbezogen werden.

Die Inhaltsanbieterinnen und -bieter sollen Zugang zu einem Computerwerkzeug für die Erstellung von Meta-daten haben.

Diese Metadaten werden später von Museumsangestellten kontrolliert und mit einer Datenbank verknüpft, bevor die Bilder online veröffentlicht werden.

Ebenfalls notwendig ist ein integriertes System für die Speicherung der anfallenden Daten.

# Modellierung – Aufgaben

## Ihre Aufgabe

Formulieren Sie einen ersten Modellversuch für diesen Problemkomplex:

Erstellen Sie ein vorläufiges Model und beschreiben Sie den Modellierungsprozess.

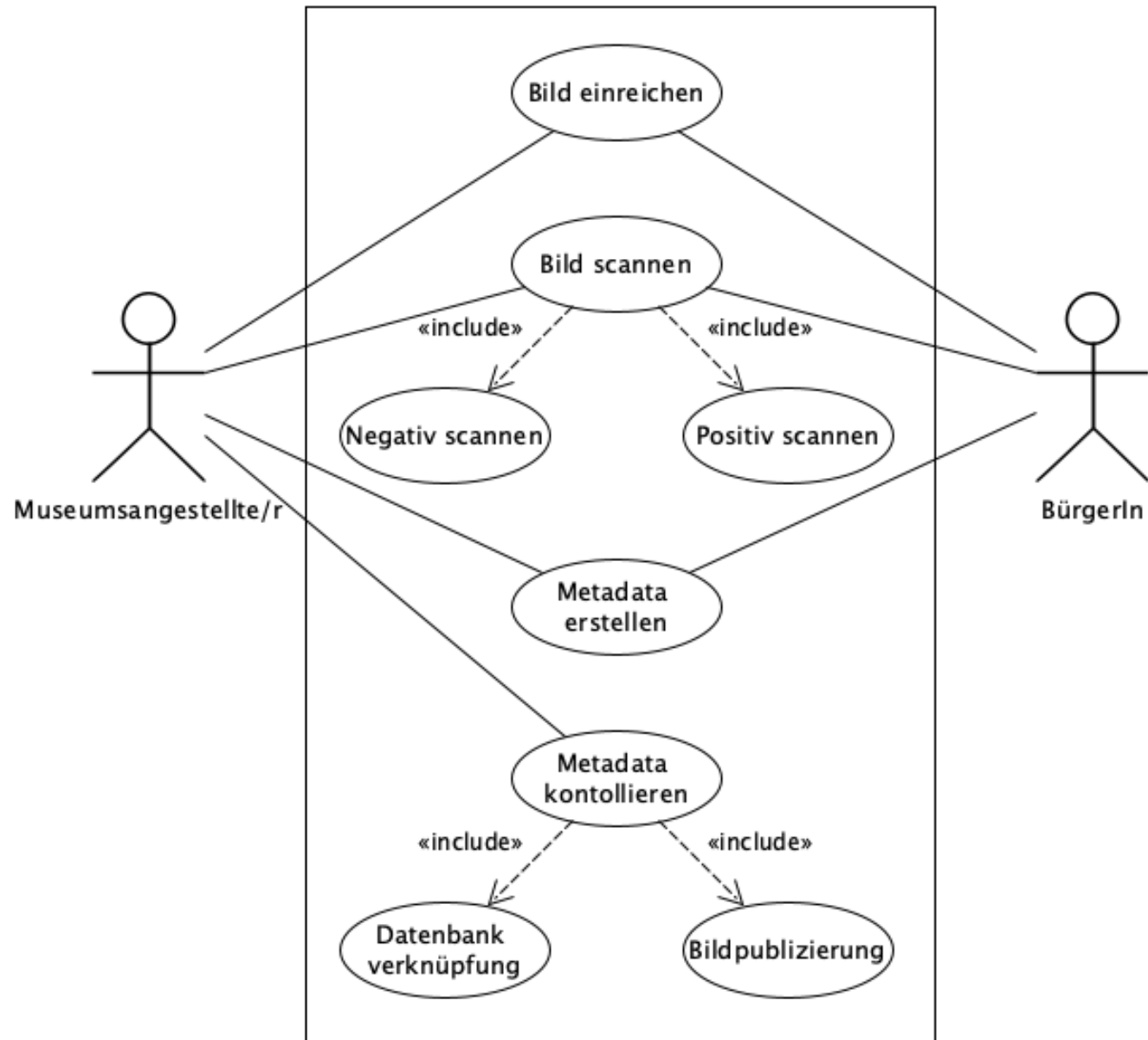
Text und UML-Diagramme zusammen mit textuelle Beschreibungen.

*Einreichung: 7. Mai 17:00*

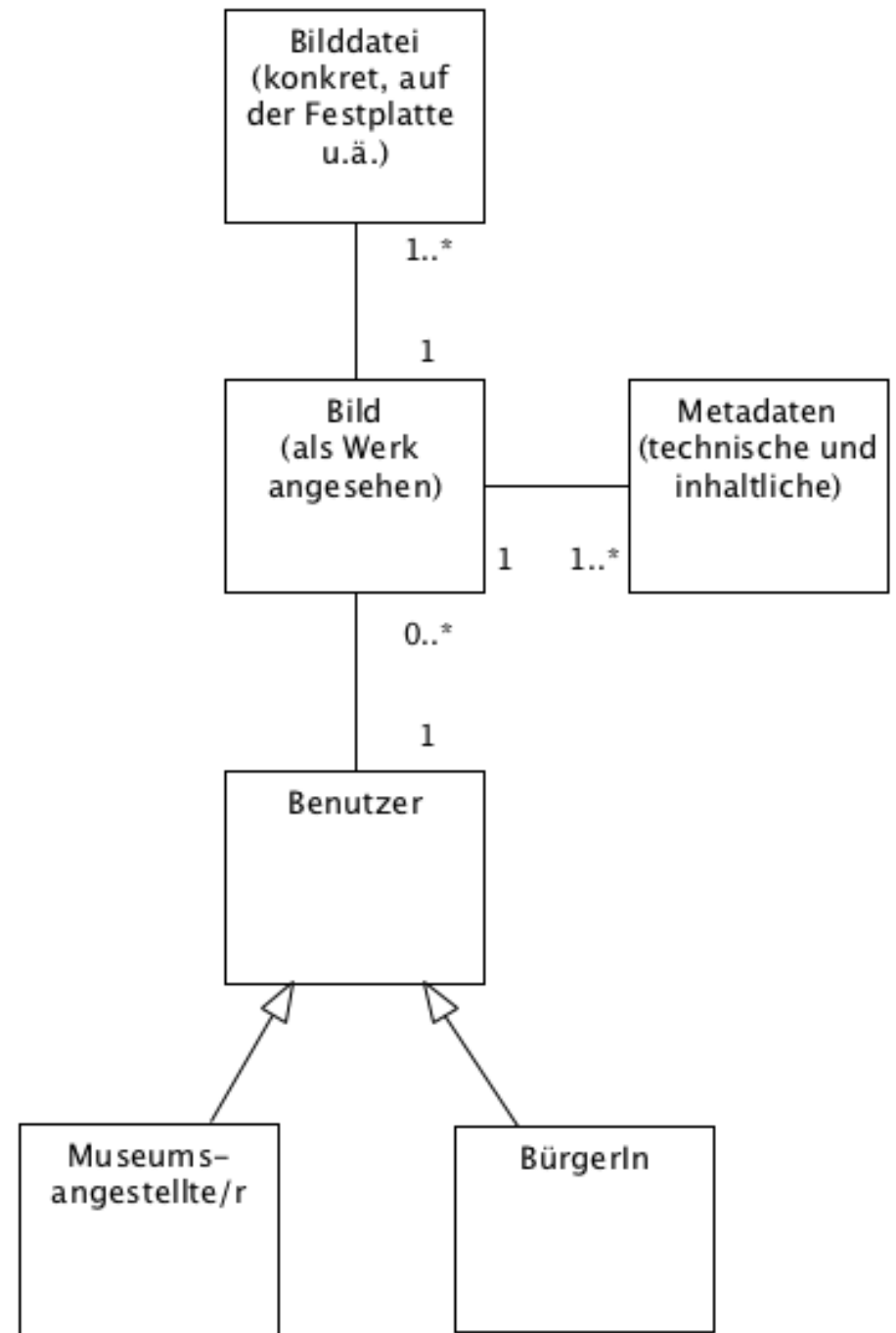
*Kommentaren in der Vorlesung 9. Mai*



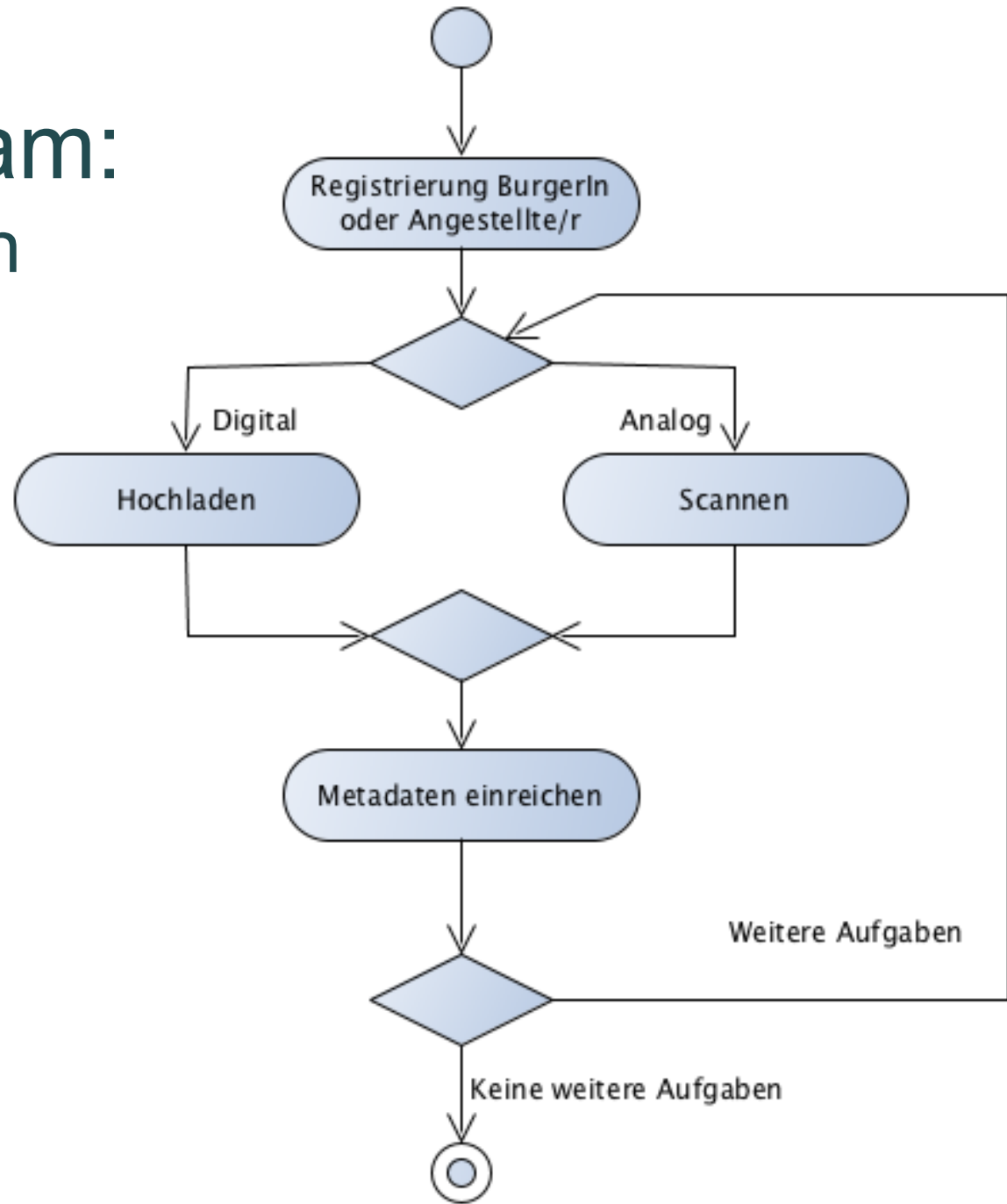
# Use case diagram



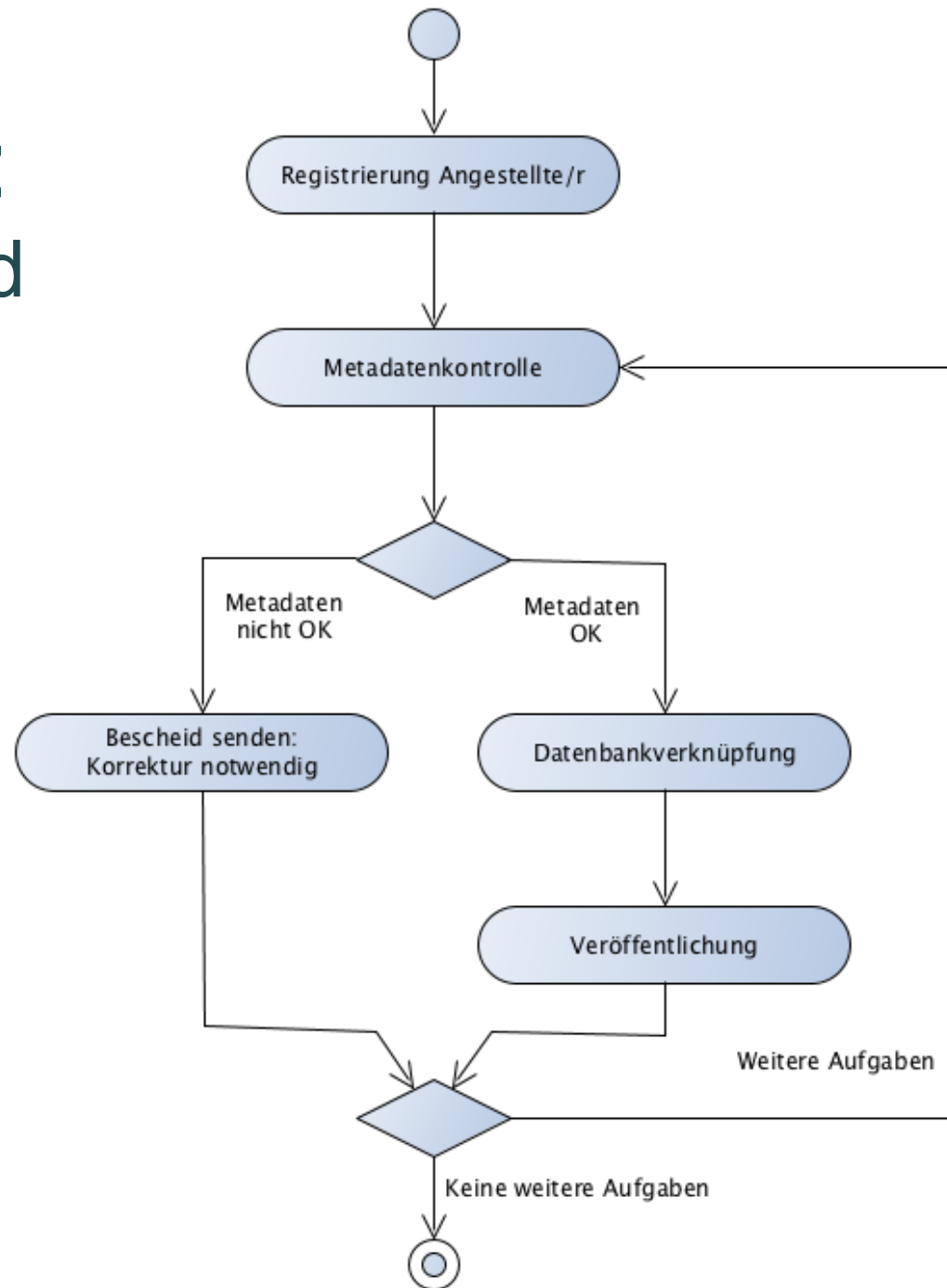
# Class diagram



# Activity diagram: enter images with metadata



# Activity diagram: metadata control and image publication





# UML

- Unification of different languages
  - the main object oriented modelling tools
- Scope more than modelling software systems
  - also systems *involving* software
  - software intensive systems
  - also beyond software intensive systems



# UML

- Use case diagrams and actors
  - actors are roles
  - benefit to a primary actor (e.g. customer), otherwise no point
  - secondary actors do the job
  - extend: decisions by stakeholders leading to actions
  - use case gives overview, more details in other diagrams
- Why standards
  - common understanding
  - UML and CRM
- Using UML tools to find problems in your modelling
  - inconsistencies between the different diagrams



# Object Management Group (OMG)

- A modelling language is a formal language for writing models
  - A programming language is a modelling language for computation
- OMG provides modelling language standards for:
  - modelling software designs
  - modelling system architectures
  - modelling business processes
  - ...



# OMG family

- UML profiles
- The UML family of standards
  - *The CRM family*
- Concrete extensions and abstract formalisations
- fUML: The Foundational UML Subset
  - an executable subset of standard UML
  - can be used to define the structural and behavioural semantics of systems
  - can be used to define MOF-based modelling languages
  - *simulations*



# OMG family

- Alf: Action Language for Foundational UML
  - extended notation, can represent structural modelling elements
  - a UML model can be represented entirely
- PSCS
  - an extension of fUML syntax and semantics
  - account for structural and behavioural semantics implied by the syntactic extensions
  - *precise semantics*
  - enable modelling and execution of UML composite structures
  - *composite structure: a UML structural diagram that provides a logical overview of all or part of a software system*



# MOF: Meta-Object Facility

- MOF is designed as a four-layered architecture
- A meta-meta model at the top layer: the M3 layer
  - the language used by MOF to build metamodels
- Metamodels: M2-models
  - most important: the UML metamodel
  - the model that describes the UML itself
  - M2-models describe elements of the M1-layer
- M1-models
  - models written in UML
- M0-layer
  - data layer
  - describe real-world objects



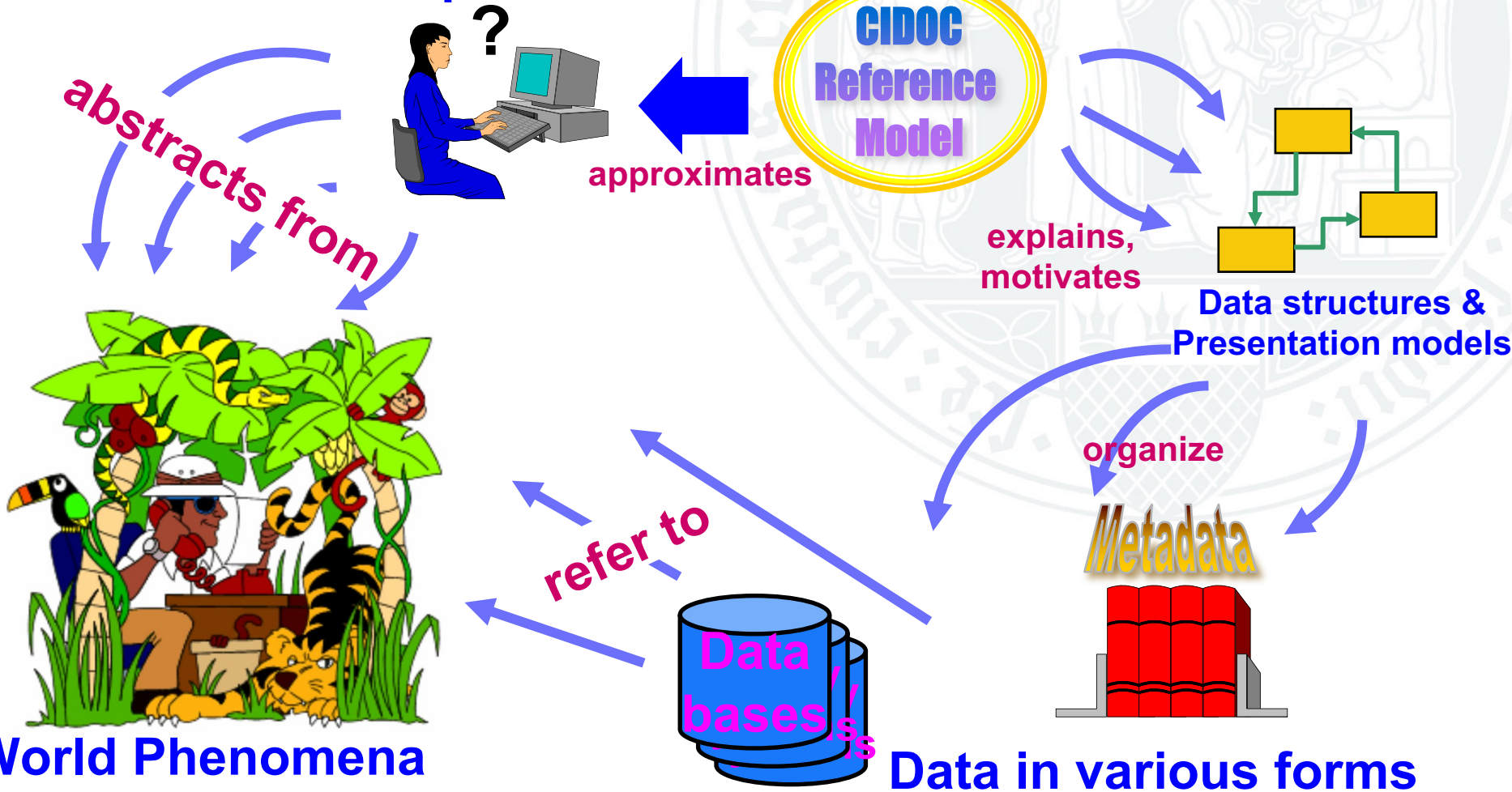
# MOF and CORBA

- MOF: create and manipulate models and metamodels
  - *CORBA interfaces*
- CORBA: Common Object Request Broker Architecture
- Facilitate the communication of systems that are deployed on diverse platforms
- Enables collaboration between systems on different
  - operating systems
  - programming languages
  - computing hardware
- Uses an object-oriented model
  - systems that use the CORBA do not have to be object-oriented
- Distributed object paradigm



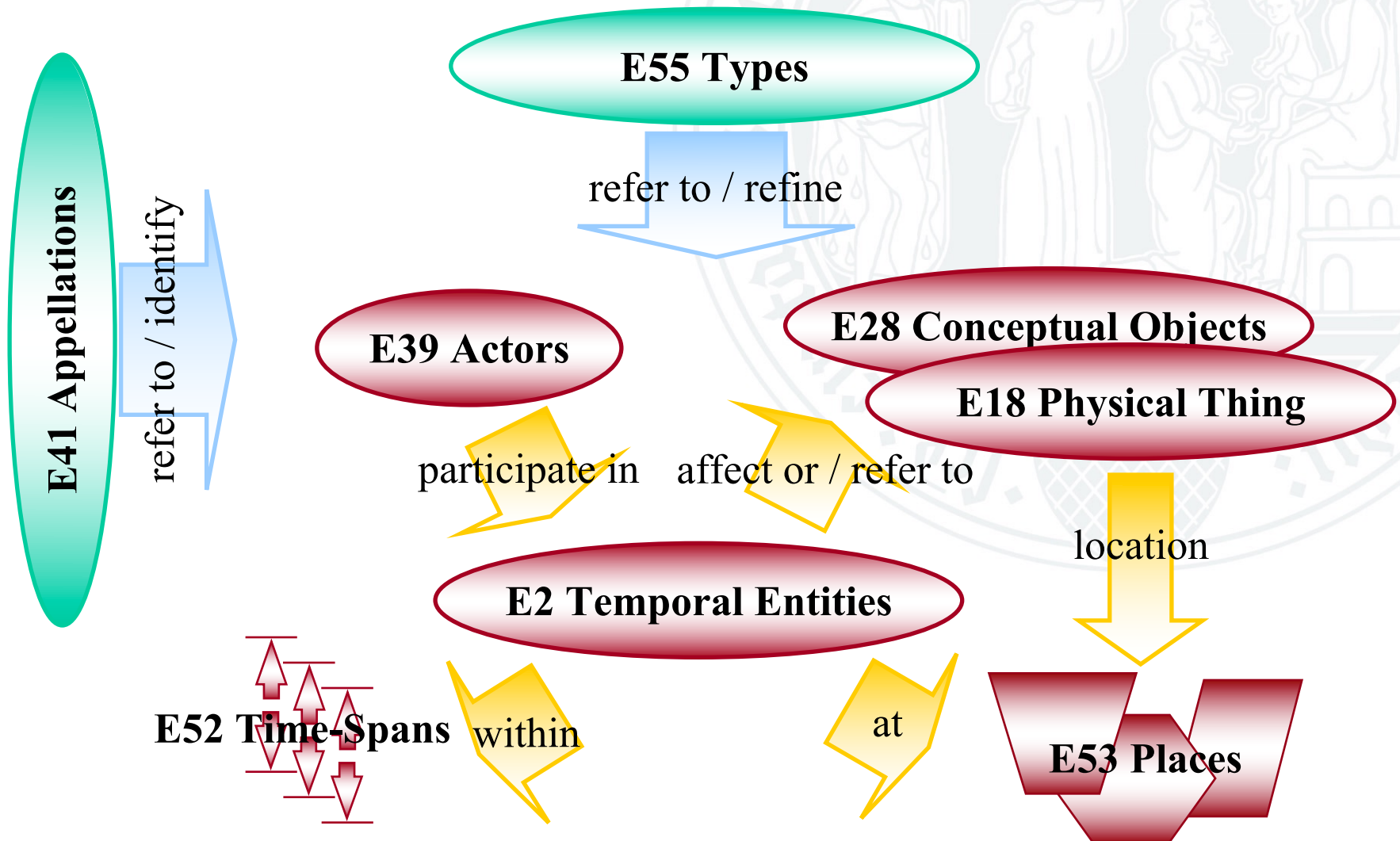
# The Intellectual Role of the CRM

## Conceptualization





# Top-level classes useful for integration



# Cultural heritage reality

*Heterogeneous sources of cultural heritage information*

CRM abstract and extensible  
→ abstraction

*Underlying semantics*



# CRM objectives

- Enable information exchange
- Enable integration
- Define (and clarify)
  - underlying semantics
  - structures
    - formal ontology
  - no terminology
    - E55 Type
  - not prescriptive as to what to document
    - but how to document it



# CRM objectives

- Enable semantic interoperability
- Not implementation-specific
- Functionalities
  - inform developers
  - common language
    - domain experts + IT developers (~ UML)
    - common formal language
    - associative queries (pattern matching)
    - support NLP
    - not replacing natural language



# CRM scope

*The curated knowledge of museums*

