FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

EDOC 2012, Beijing, China
14 September 2012 17:00

Automating the Management and Versioning of Service Models at Runtime to Support Service Monitoring

Ta'id HOLMES [1], Uwe ZDUN [2], Schahram DUSTDAR [1]

[1] Distributed Systems Group, Institute of Information Systems
Vienna University of Technology, Austria

[2] Software Architecture Group, Faculty of Computer Science
University of Vienna, Austria

COMPAS
SEVENTH FRAMEWORK PROGRAMME
VITALAB Distributed Systems Group

Context and Background

software systems become increasingly **complex**

- unify different technologies
- are adapted for new and emerging technologies
- need to comply with imposing requirements

Model-Driven Engineering (MDE)

- helps to master complexity (design-time)
- utilizes **models** as central artifacts

MDE — Problems

- evolution and co-evolution of MDE artifacts and systems
- concurrent work
  - few MDE tools offer collaboration support ⇒ lack of integration
  - common version control systems are too naïve for MDE
    - ⇒ versioning on a model element level is not supported
    - ⇒ relationships between artifacts are not captured/managed
- search & retrieval of models and MDE artifacts
  - missing tool support and infrastructures ⇒ reuse becomes difficult
- traceability (high-level ↔ low-level model-instances and code)
  - essential for meaningful feedback from the runtime to stakeholders and for identifying and understanding the root-cause
- generation step causes different MDE phases to be isolated
  - missing infrastructure that supports dynamic model look-up ⇒ model-based reflection is rarely used
- monitoring of model-driven systems (e.g., in regard to requirements)

Addressing MDE Challenges

- Support for *various* stakeholders
  - appropriate model-representations (DSLs)
  - role-based access controls (RBACs)
- Dealing with *concurrency*
  - locking mechanisms
  - raising the awareness of the work of others
  - comparing and merging possibilities
  - support for resolving conflicts
- Management of *MDE Projects & Artifacts*
  - versioning
  - capturing and keeping track of relationships
  - support for model evolution
- Support for model-aware *entities*
  - information retrieval services
  - resource management services

1. **Mo**del-Aware

   because it stores models & MDD artifacts

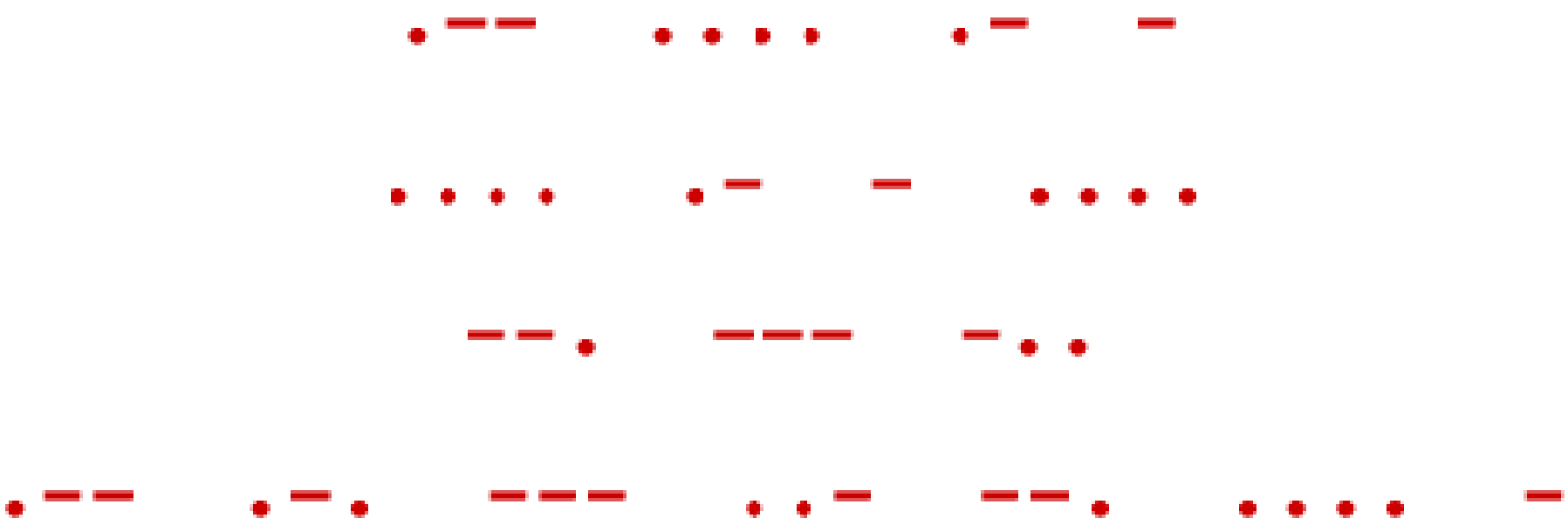
2. **R**epository

   because it supports configuration management (e.g., versioning) of MDD projects

3. & **S**ervice **E**nvironment

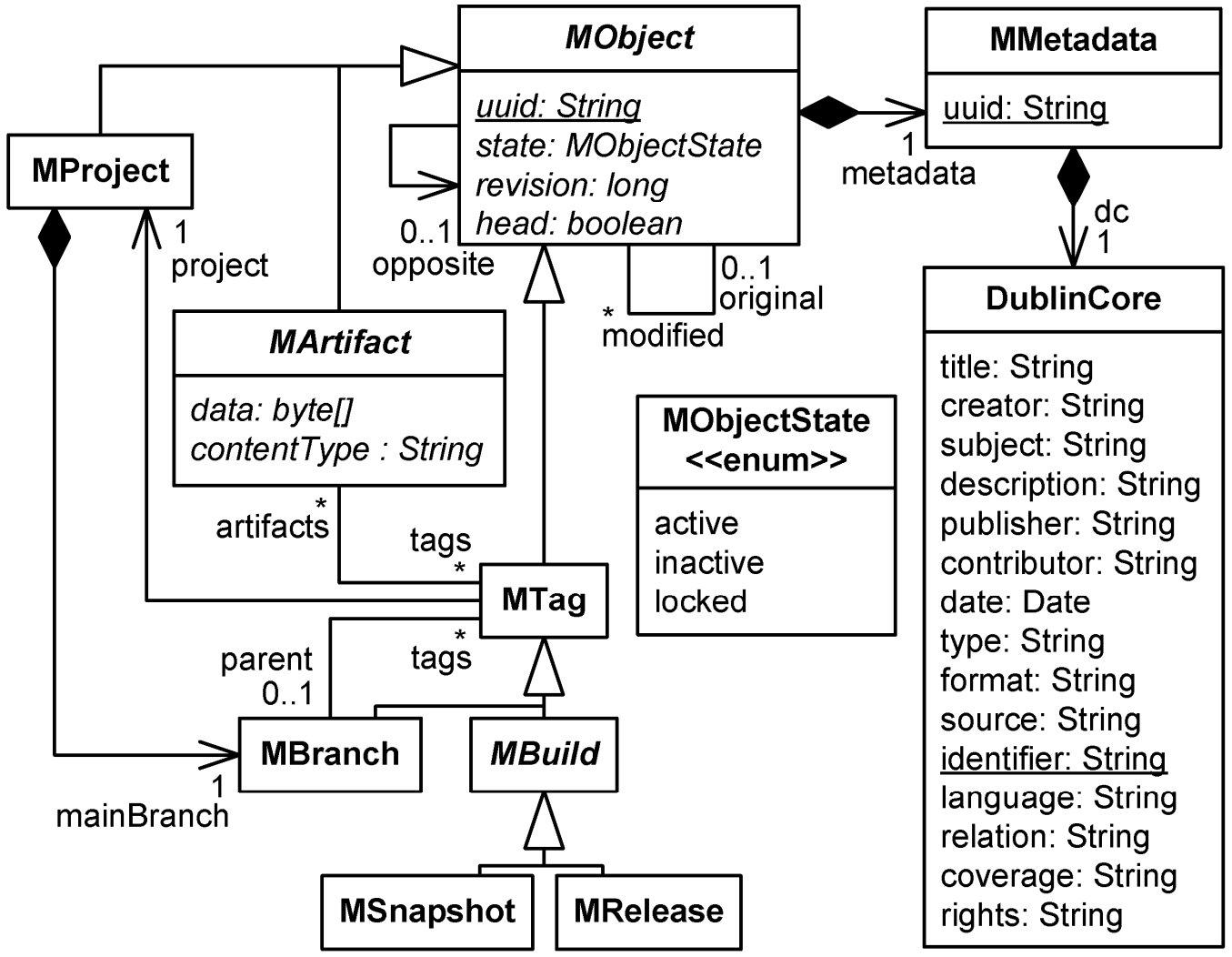
   because it offers service-based interfaces and integrates with other model-aware components, that cover the model-driven engineering lifecycle

The First Electronic Message



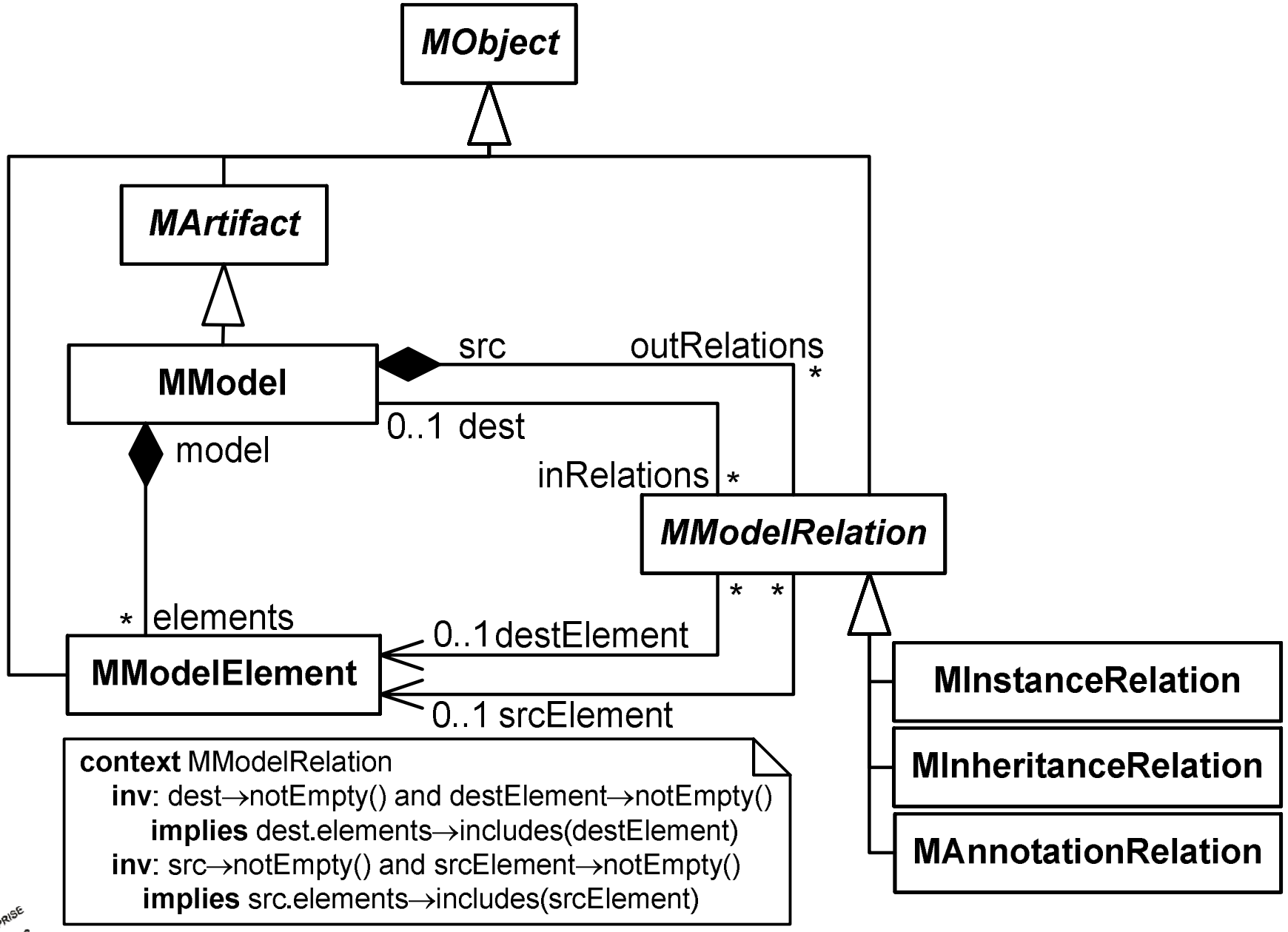
Samuel F. B. Morse
May 24th, 1844

MORSE Core Model



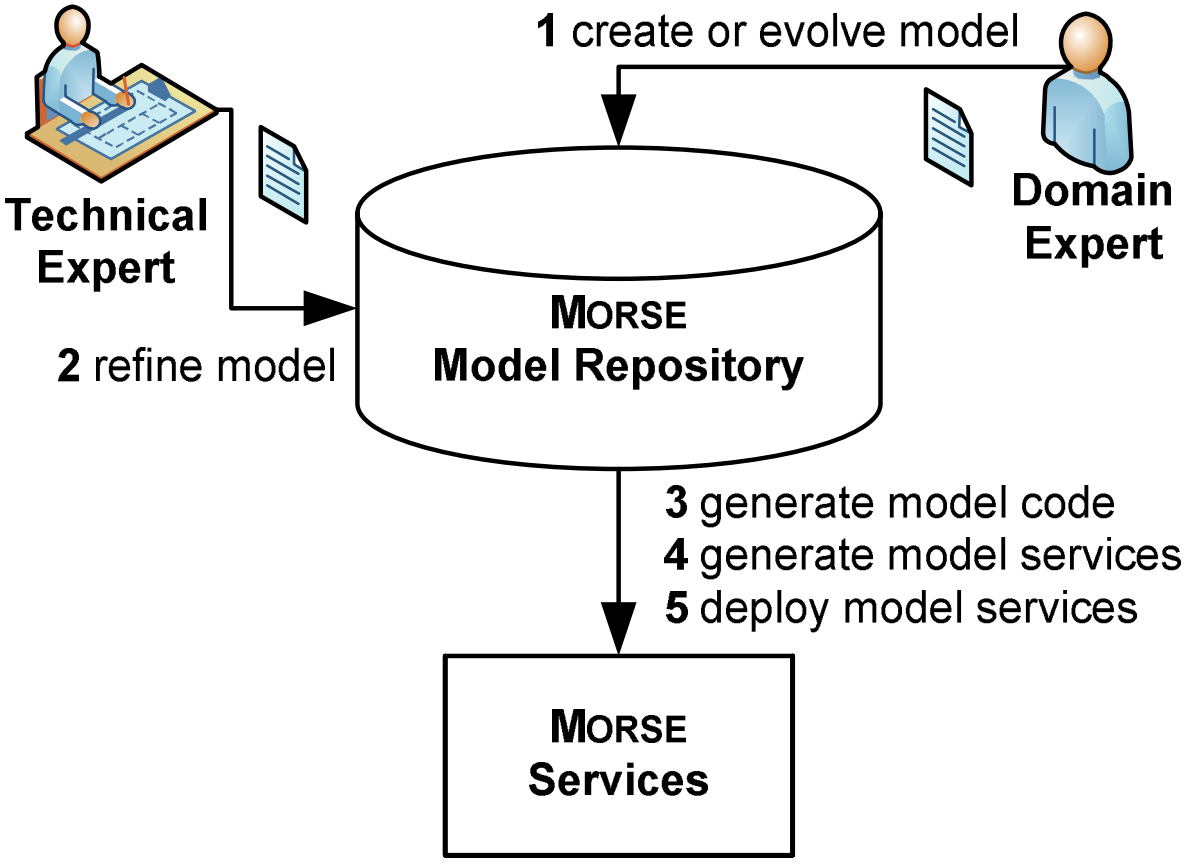

```
context MObject
   inv: metadata.uuid = uuid and metadata.dc.identifier = uuid
   inv: opposite→notEmpty() implies ((head xor opposite.head) and opposite.oclIsTypeOf(self))
   inv: original→notEmpty() implies original.oclIsTypeOf(self)
context MProject inv: mainBranch.project = self
context MTag inv: parent→notEmpty() implies parent.project = project
```

MModel, -Element, -Relation



MORSE Services

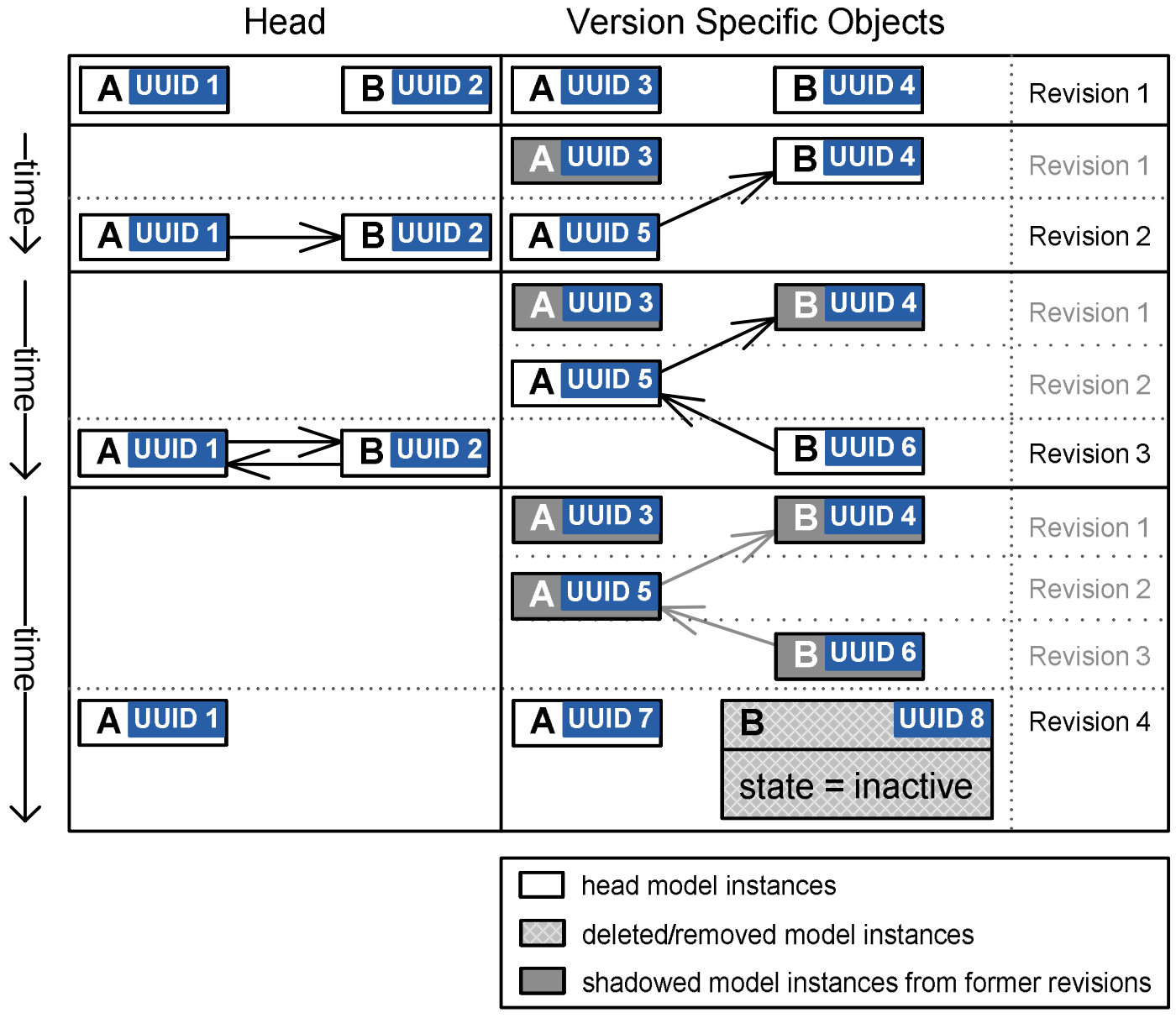
- generated for each concept of a model
  - information retrieval
  - resource management
  - versioning



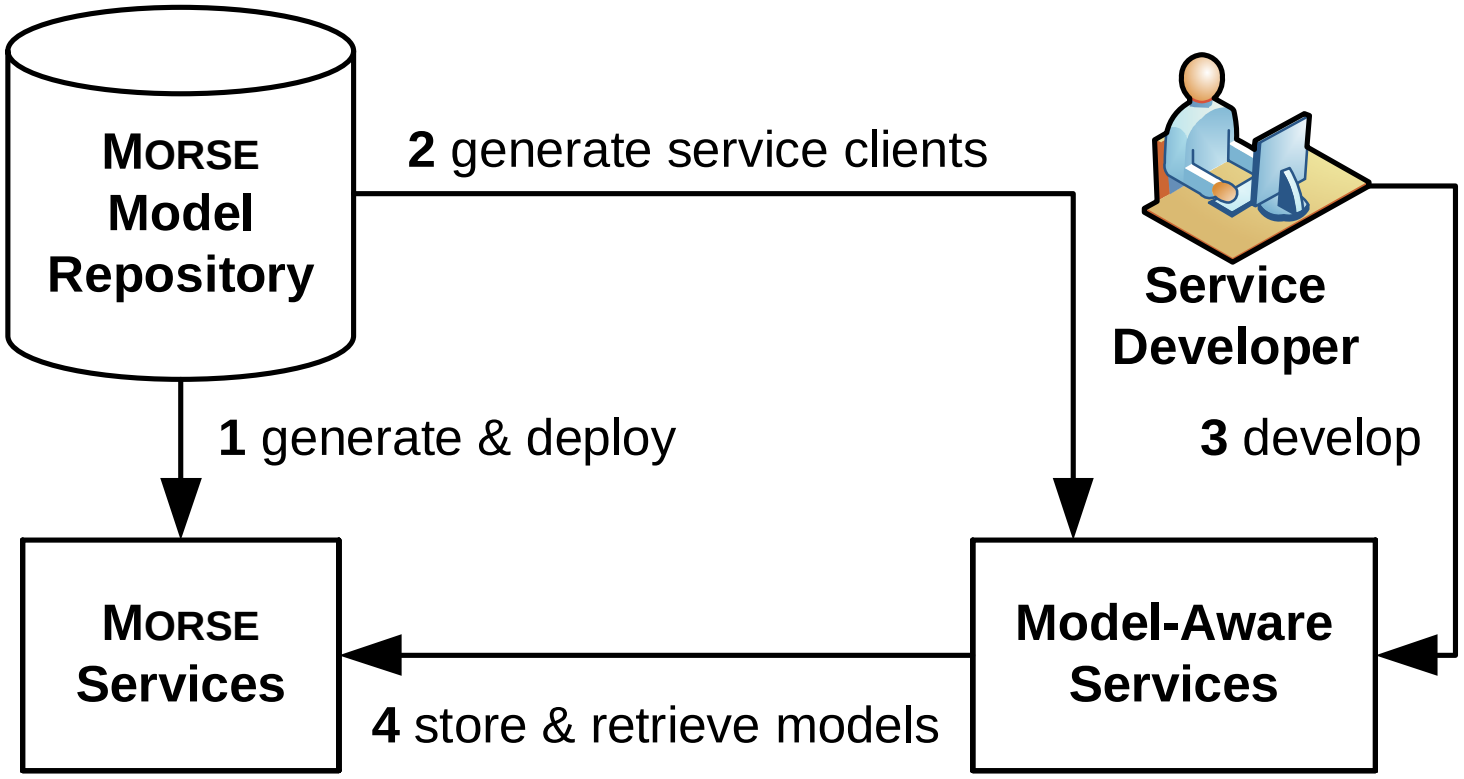
MORSE Service Operations

Response	Operation	Description
boolean	**exists**	does a model with a UUID exist?
boolean	**isHead**	is the object (specified by UUID) version-independent?
UUID[]	**list**	returns the VIIDs of all models
UUID[]	**versions**	returns all VSIDs of a model
<Class>[]	**query**	search for models; support of various query parameters
<Class>	**retrieve**	a model is retrieved by UUID
UUID	**create**	a VIID is returned
UUID	**update**	a VSID is returned
UUID	**delete**	a VSID is returned
UUID[]	**list<Role>**	returns the UUIDs for a role
UUID	**add<Role>**	a VSID is returned
UUID	**remove<Role>**	a VSID is returned
UUID	**clear<Role>**	a VSID is returned

UUID-Based Model Versioning



MORSE Service Clients



Model Repository Comparison

Repository	Model Identification	Model Element Identification	Model Navigation	Complex Search
AMOR	URL	ID	✘	✘
AtlanticZoo	URL	✘	✘	✘
CDO	URL	URI-Fragment	✓	✓
EMFStore	ID	ID	✓	✘
MDR	ID	URI-Fragment	✘	✘
ModelBus	URL	✘	✘	✘
MORSE	**UUID**	**UUID**	✓	✓
Odyssey-SCM	ID	URI-Fragment	✘	✘
Odyssey-VCS 2	ID	URI-Fragment	✘	✘

Model Repository Comparison (2)

Repository	Modeling Technology	Unit of Versioning
AMOR	EMF	ANY
AtlanticZoo	ANY	model
CDO	EMF	M2 class instance
EMFStore	EMF	ANY
MDR	MOF 1.4	ANY
ModelBus	ANY	model
MORSE	ANY	M2 class instance
Odyssey-SCM	MOF 1.4	ANY
Odyssey-VCS 2	EMF	ANY

Reproducibility

- Project Website:
  http://www.infosys.tuwien.ac.at/prototype/morse
- Online Documentation:
  https://www.infosys.tuwien.ac.at/m2projects/
- Libraries retrievable from Maven-Repository:
  https://www.infosys.tuwien.ac.at/maven/
- Subversion Repository:
  https://svn.vitalab.tuwien.ac.at/projects/morse/
- Open Source:
  Apache License, Version 2

Scientific Contributions

- We present a *novel* transparent UUID-based model versioning technique.
- In a service-oriented environment we make MDE artifacts uniquely identifiable and automate the generation of retrieval and management services.
- We unify the use and management of MDE projects and artifacts for design time and runtime clients.

Summary

- first model repository based approach to support models and **model reflection** at runtime in SOC
  - the usability of models at runtime is enhanced through the automated model-driven generation of specialized traversal, query, and storage services
- an **evolution of models**
  - services that rely on models in distinct versions can easily relate to these due to UUID-based model versioning

Thanks for your attention!
谢谢!

Ta'id HOLMES (福尔摩斯·大山)
Distributed Systems Group
Institute of Information Systems
TU Wien

http://www.infosys.tuwien.ac.at