# Rule driven workflow enactment

Adriaan van Os

Tim Verwaart

www.soops.nl

www.lei.wageningen-ur.nl

ESUG 2005
Brussels

# Subject

- Last year we presented Artis: a system for model driven development and deployment
- In Artis, workflow is driven by the information need
- Data models and business rules can be used to specify the information need
- In this presentation we explain how we realized workflow enactment by comparing database content with business rules

# Workflow enactment

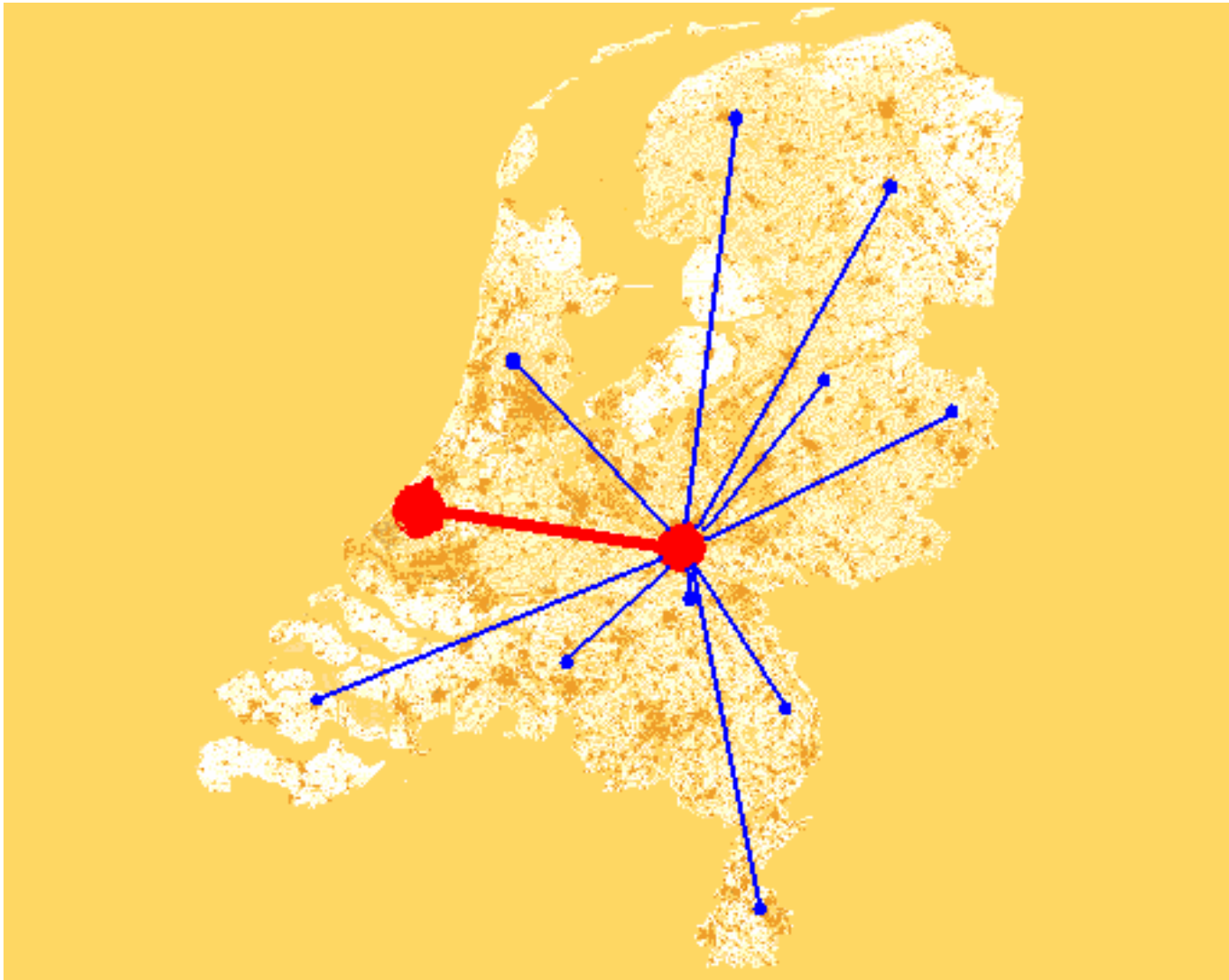Rule violation entered into workload for responsible employee
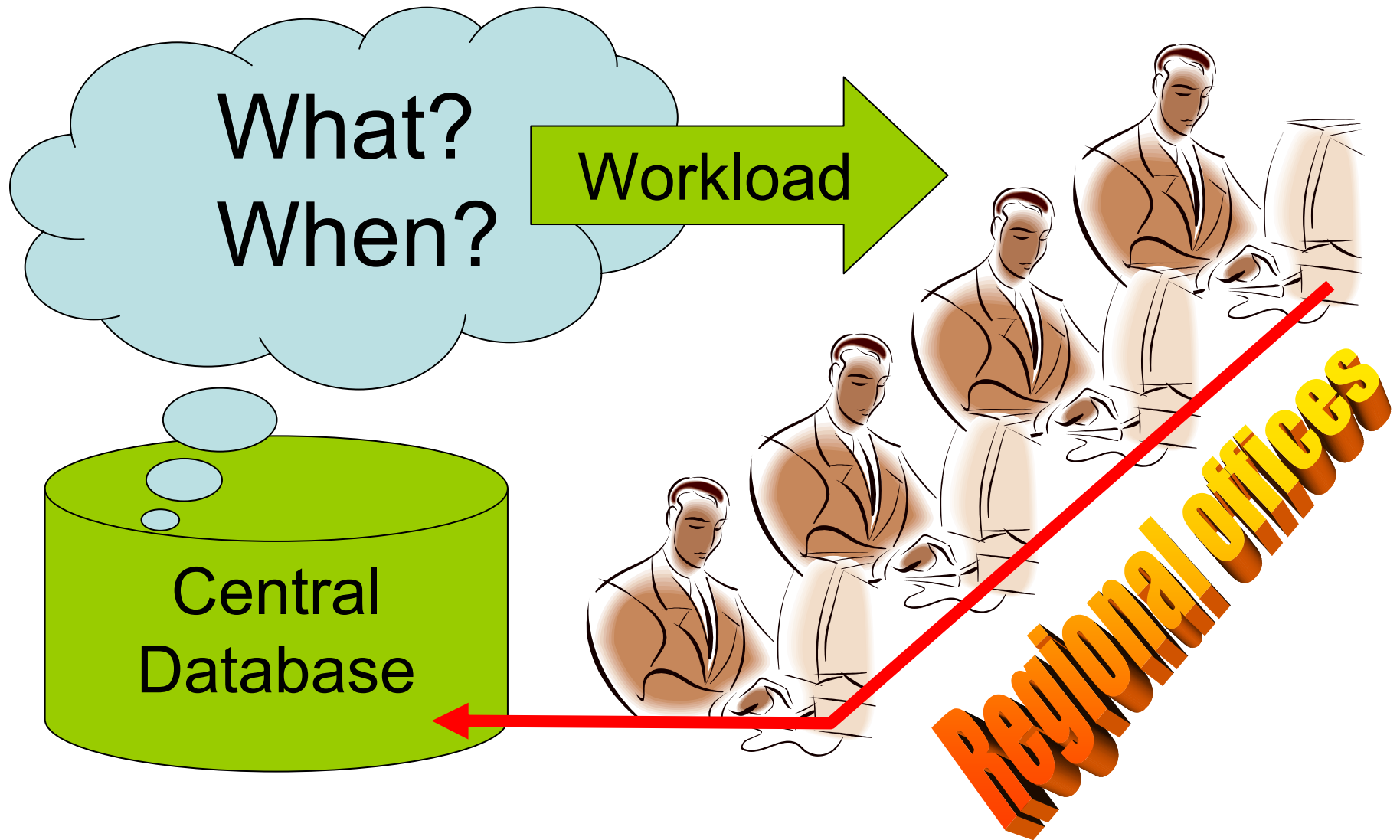


Enact procedure

# Information need
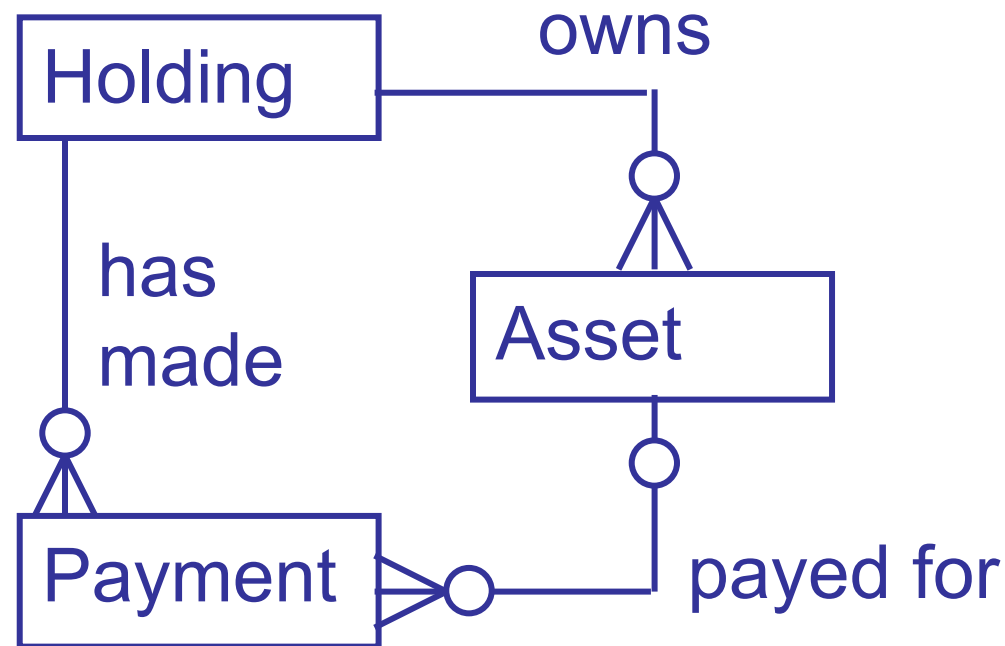
# In our case: farming

# Regional offices

# Information need drives workflow

# Information need can be represented as a data structure model, for instance (attributes not shown):

# Data model is not sufficient to define the information need

- We do not need to know all attribute values in all cases
- We want the data to satisfy integrity constraints
- We need the information in time

**relevant**

**complete+**

**consistent**

**current**

# Attach rules to each attribute

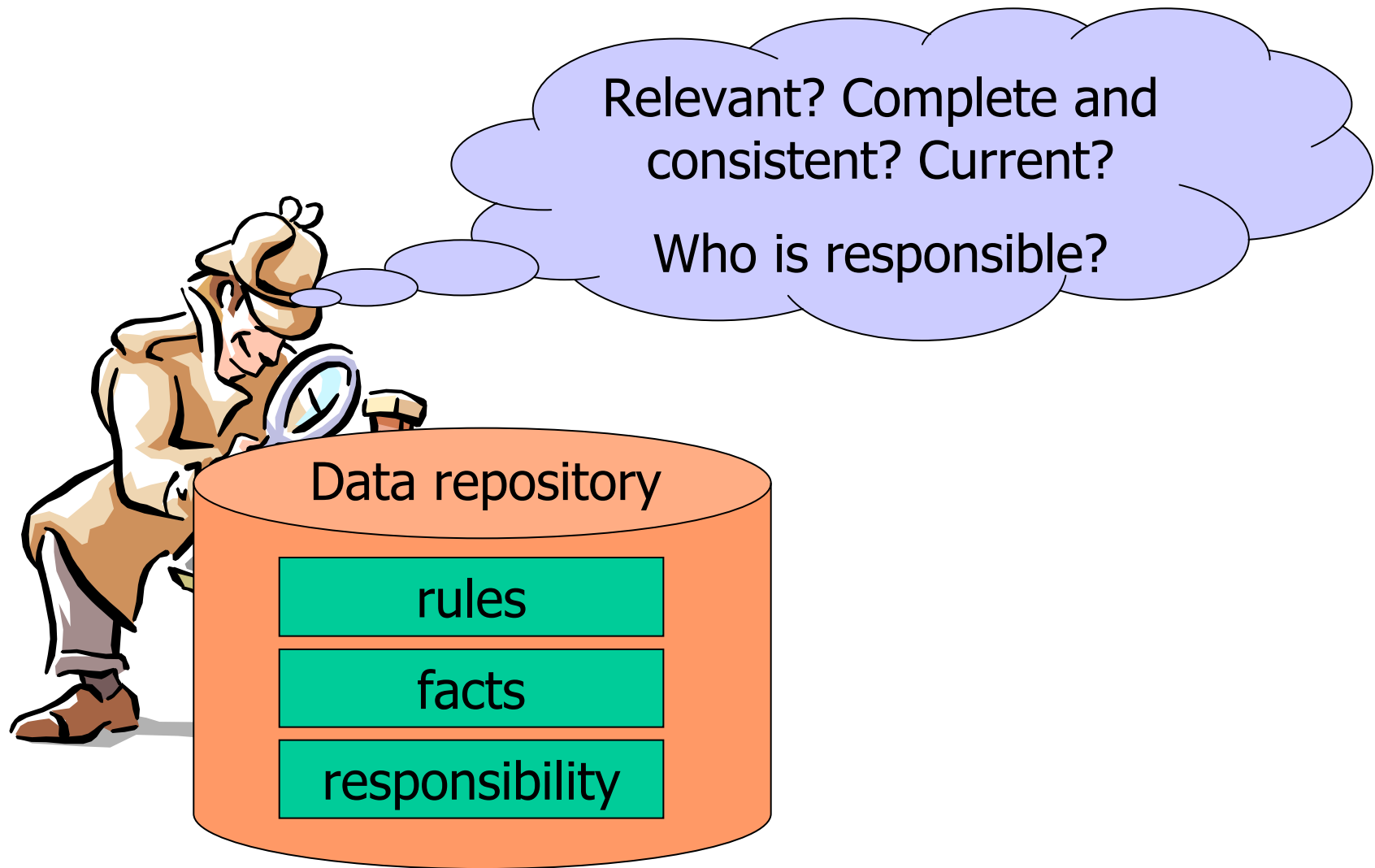| | |
|---|---|
| **relevant** | **Relevance rule** |
| **complete+ consistent** | **Integrity rule** |
| **current** | **Actuality rule** |

Demo: RIA rule formulation and representation

# Act in case of rule violations

# Who is responsible?
# What should she do?

# Keeper determines workload

# Keeper algorithm

- for each entity in some designated population as a starting point:
  - for each attribute:
    - determine responsible employee and add rule violation to the workload:
      - if the attribute is relevant and the value is not current
      - or if the attribute is relevant and the value is current but does not satisfy the integrity constraint
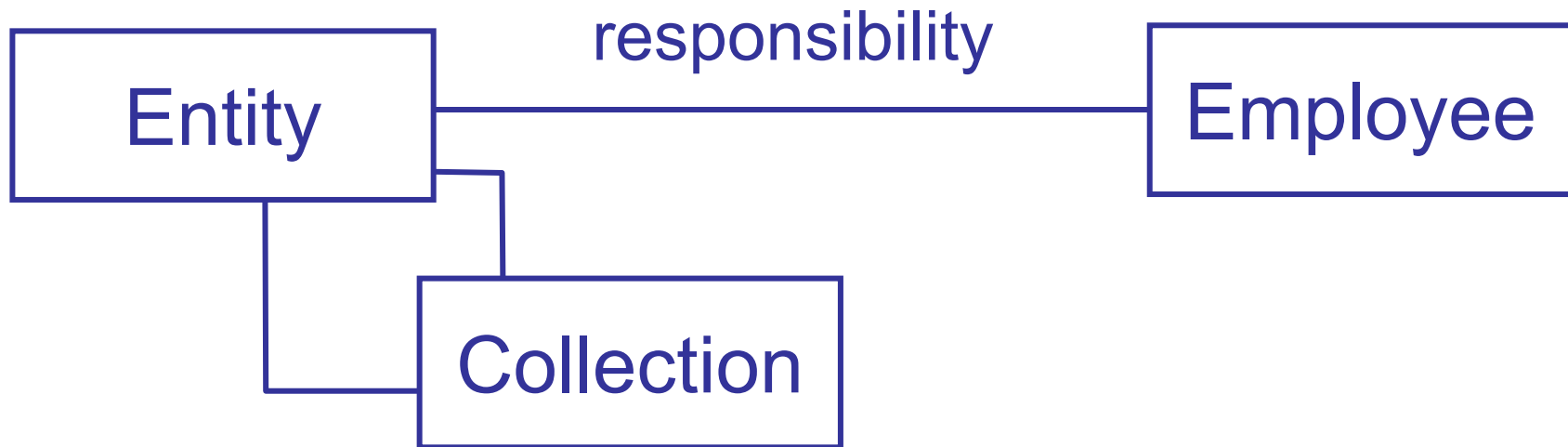
# Two remarks

1. Attributes may be multi-valued (have a collection of entities as value). In that case, execute the keeper algorithm for all entities in the collection too.

2. Evaluation of actuality rules requires a temporal data representation. We represent the value of facts not as a simple value but as a pair of value and validity time.

Demo: data representation (viewer)

# Who is responsible?

- We record responsibility for entities in the repository as an entity-employee-association

- No need to designate responsibility for each entity. The keeper traces back to the entity in the population it used as starting point.

```
                  responsibility
┌──────────────┐                    ┌──────────────┐
│              │────────────────────│              │
│   Entity     │                    │   Employee   │
│              │┐                   │              │
└──────────────┘│                   └──────────────┘
         │      │
         │   ┌──┴───────────────┐
         │   │                  │
         └───│   Collection     │
             │                  │
             └──────────────────┘
```

# What shoud she do?

- The workflow is described in procedures
- We record applicability of procedures as an association of procedure and rule

applicability

| Procedure | | Rule |

- We use this knowledge to "enact" workflow: enter first task of procedure into responsible employees to-do list

Demo: workload (Mole) and start

# Conclusion

- Workflow can be enacted by comparing database content with business rules

- This can only be generically achieved by applying a temporal data model

- We need knowledge about:
  - responsibility (of persons)
  - applicability (of procedures)