

# DESIGNING METHODS FOR QUALITY

**Elvira LOCURATOLO**

*IEI, Consiglio Nazionale delle Ricerche,*

*Via Moruzzi, 1 Ghezzano - San Cataldo - Pisa*

*Telephone +39 50 3152895, Fax +39 50 3152810*

*E-mail [locuratolo@iei.pi.cnr.it](mailto:locuratolo@iei.pi.cnr.it)*

# Objectives

- Metamethod
- MetaASSO
- Approach importance
- Overview of ASSO

# Metamethod

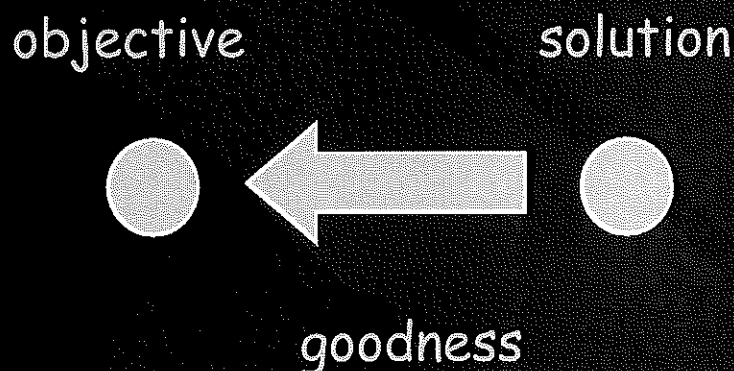
Stepwise approach to the design of methods for the achievement of quality

## Step

objective: requirement of a method

solution: proposal of a method

goodness: solution that satisfies the objective  
correctly





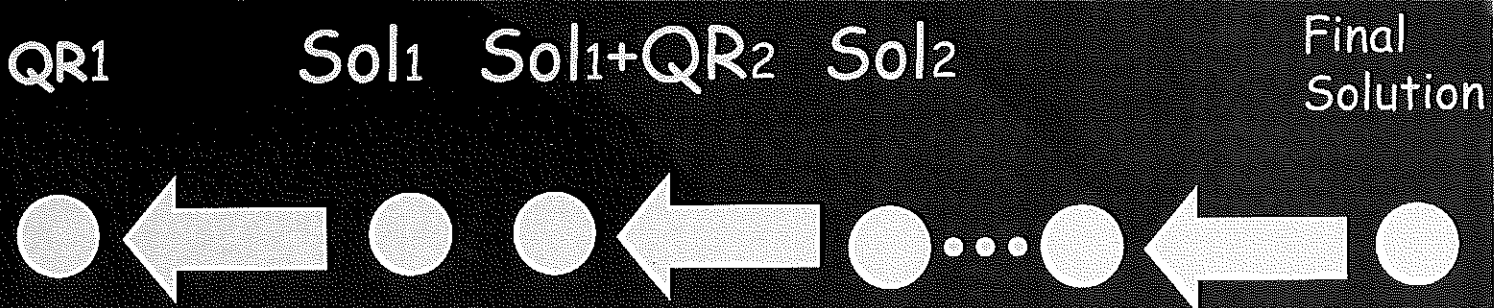
# Metamethod

## New Objective

Good solution with new quality requirements

## Quality Oriented Refinement

Solution refinement that meets the added quality requirements ensuring goodness



# MetaASSO

Metamethod example resulting into the ASSO design

## ASSO

Methodology of conceptual database design that achieves the following goal quality requirements:

- **Easiness of use**  
conceptual schema easy to use
- **Flexibility**  
conceptual schema easy to modify
- **Reliability**  
conceptual schema consistency and  
logical schema correctness
- **Economy**  
low costs of the proof process
- **Efficiency**  
limited amount of time and space

# MetaASSO: first step

## Objective:

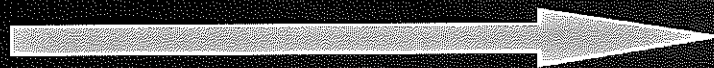
Correct proposal of a method able to conciliate flexibility with efficiency

## Solution:

Consistent  
Conceptual  
Schema



Correct Transformation



Logical  
Schema



## Goodness:

Motivations are given to guarantee that the solution satisfies the objective correctly

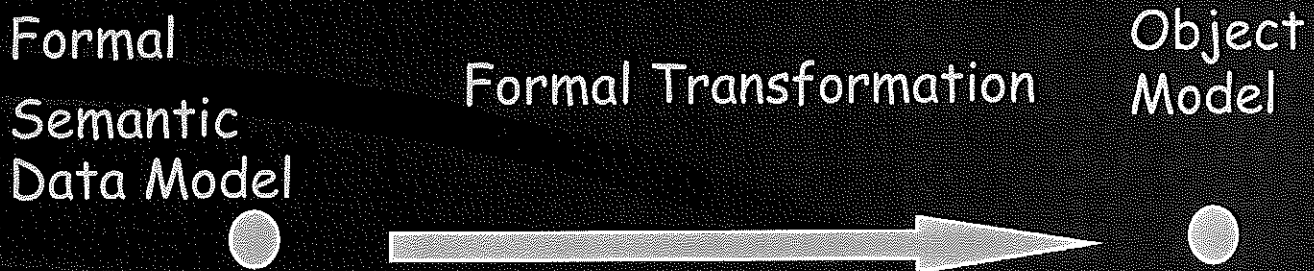


# MetaASSO: second step

## Objective:

Proposal of a consistent conceptual schema correctly linked with a logical schema.

## Solution:



## Goodness:

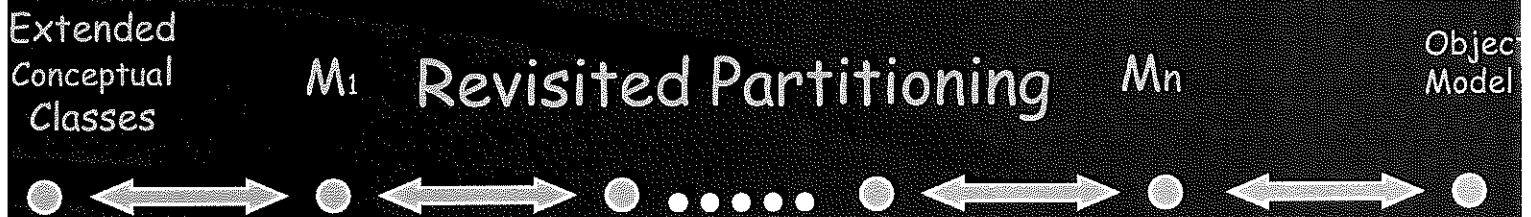
Motivations given to guarantee that the solution is a quality oriented refinement of the previous solution

# MetaASSO: third step

## Objective:

Proposal of a method to correctly transform a semantic data model into an object model.

## Solution:



## Goodness:

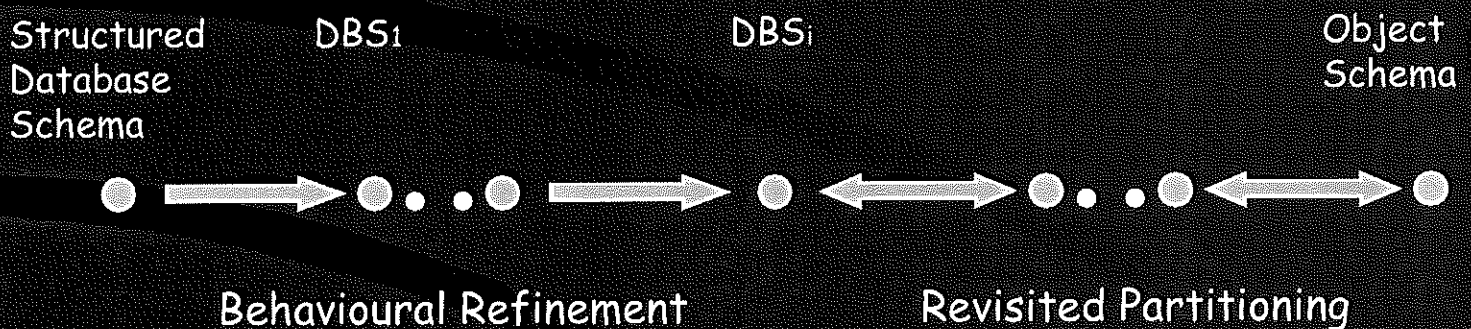
Motivations given to justify the Revisited Partitioning and the extended conceptual classes



# MetaASSO: final step

**Objective:** enrichment of the previous model with application constraints in order to still allow the Revisited Partitioning decompositions

## Solution:



## Goodness:

Formal relation between ASSO and B

# Metamethod Importance

New notion of quality in conceptual database design

## Practical benefits

- Proposals of good solutions
- Possibility to agree a new solution
- Possibility to reuse previous experience

# MetaASSO Importance

⇒ modular design of ASSO

Revisited Partitioning

Structured Database Schema

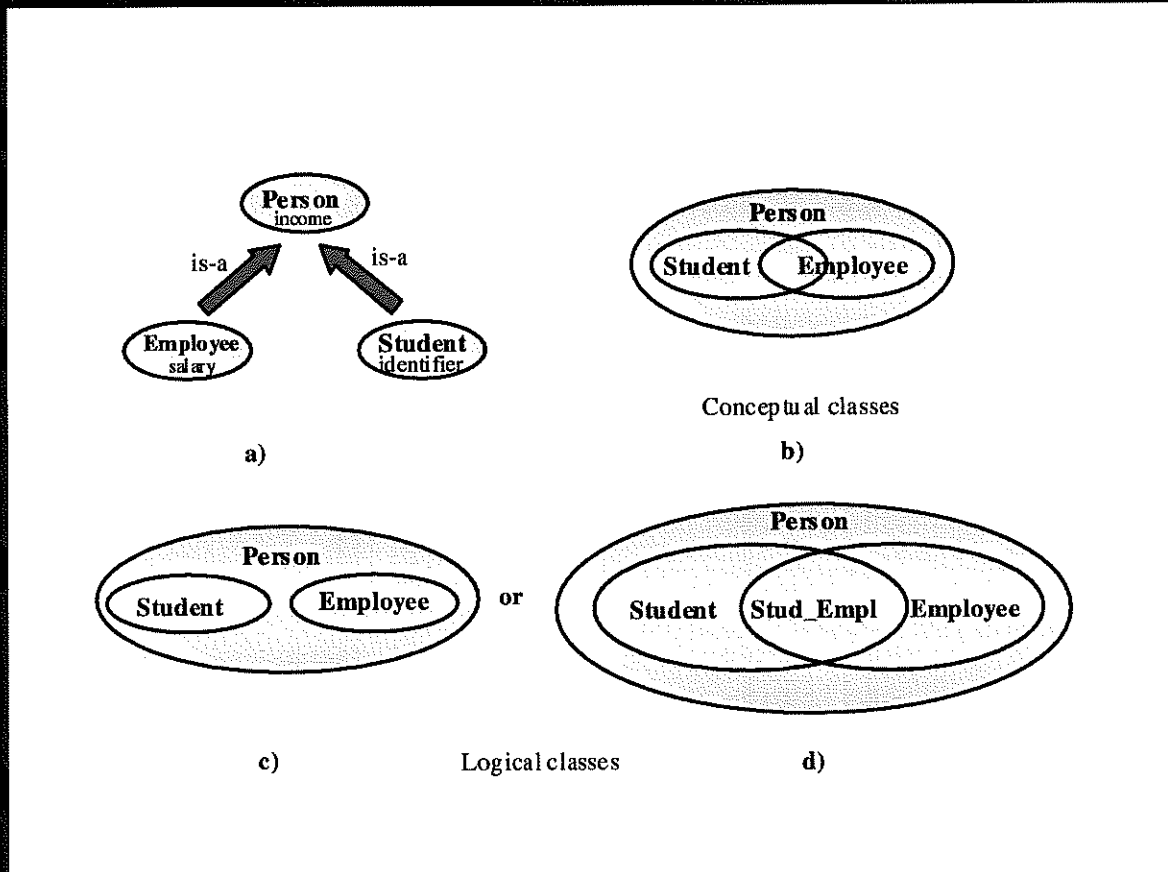
Relation between ASSO and B

⇒ reduction of the proof process

⇒ effects on the ASSO-toolkit



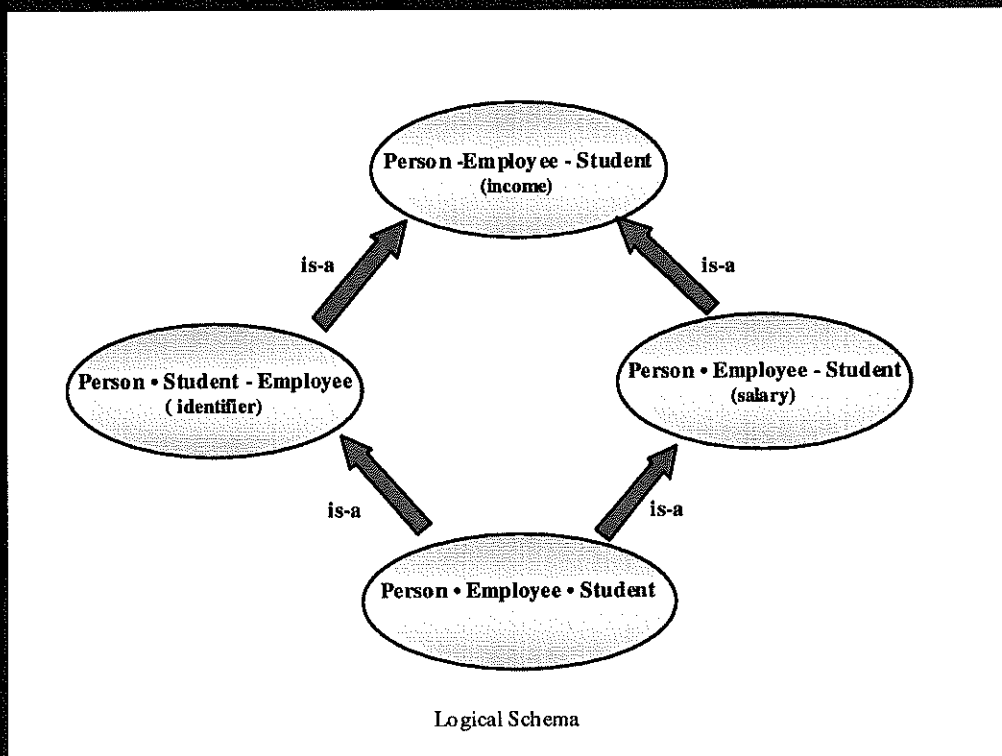
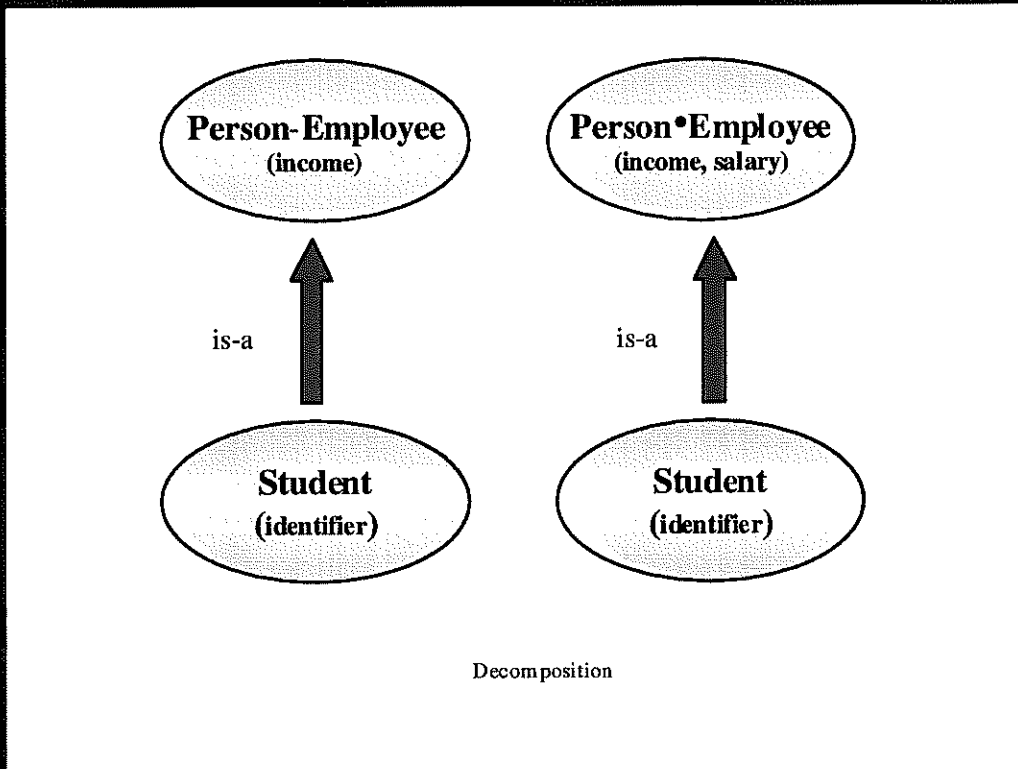
# Revisited Partitioning



Conceptual classes: each object instance can belong to any classes of a specialisation hierarchy

Object classes: each object instance belongs to one and only one classes of the specialisation hierarchy

# Revisited Partitioning

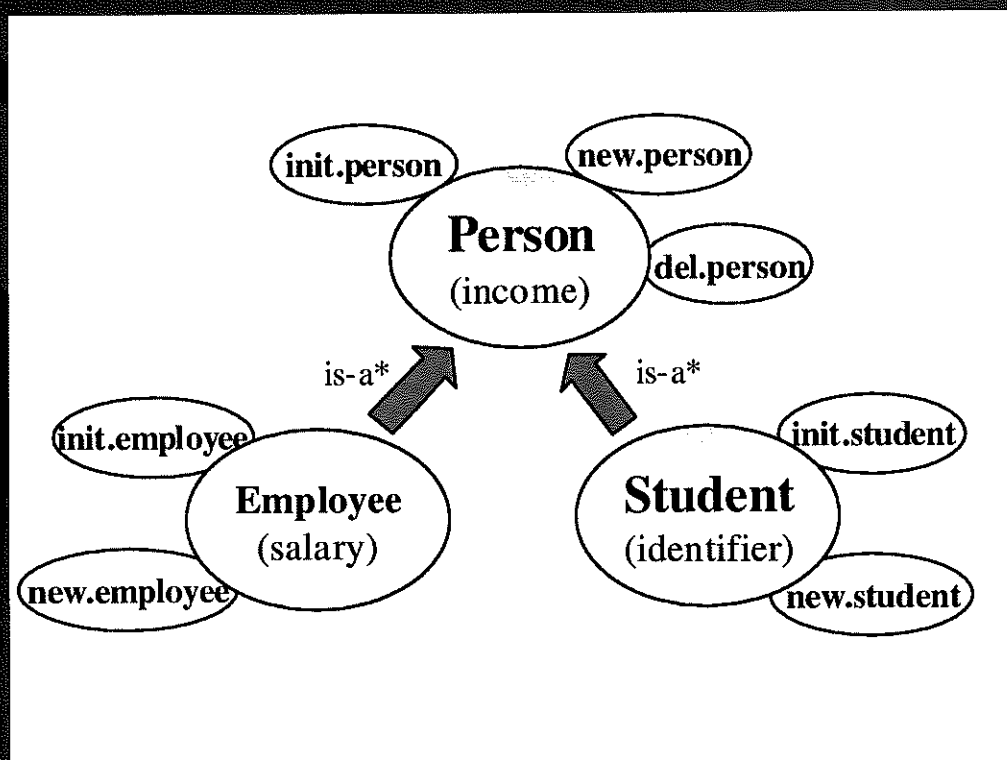


# ASSO model: first view

**Class:** mechanism able to model a set of database objects with attributes, application constrains and operation, enclosing an initialisation

**Is-a\* relationship:** is-a relationship extended to model a behavioural specialisation

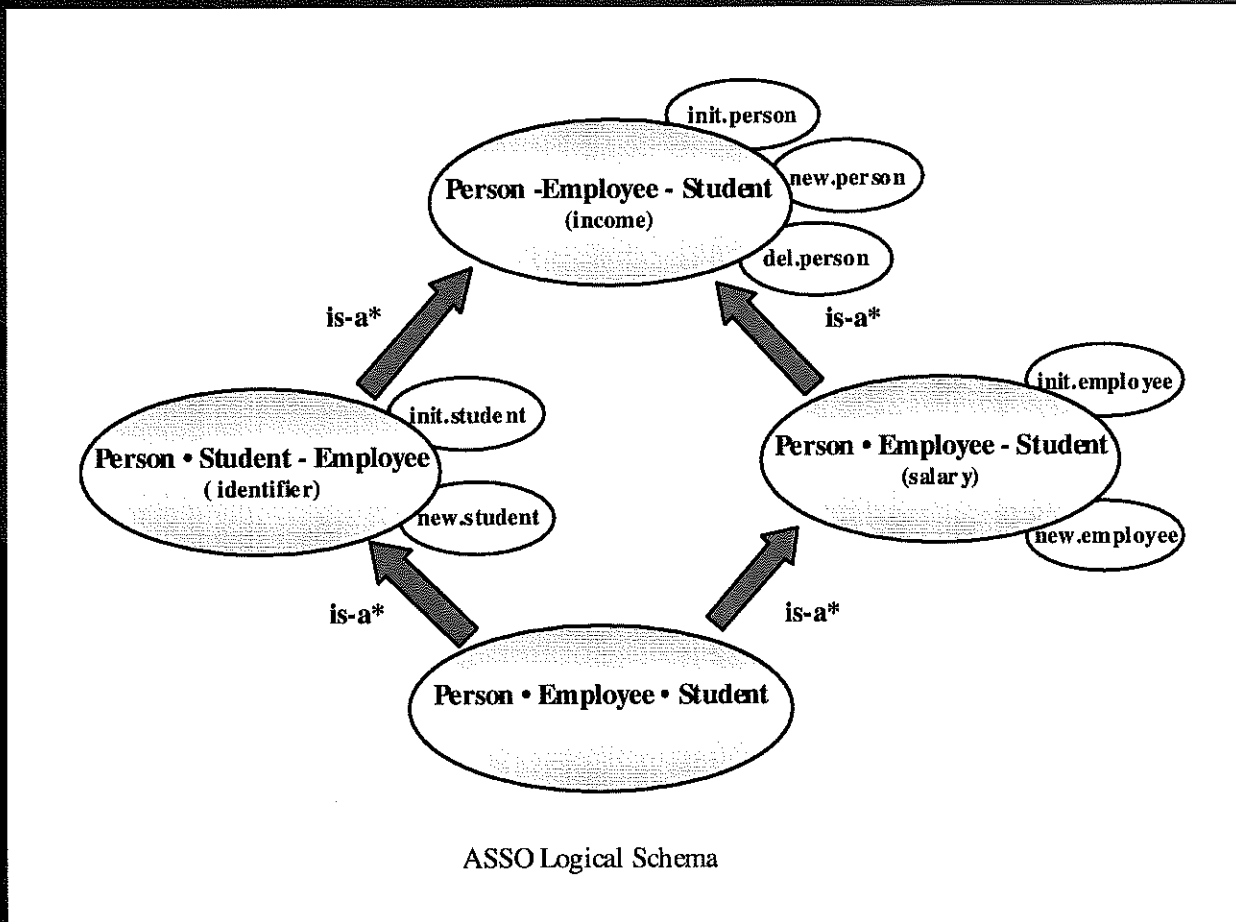
**Structured Database Schema:** connected acyclic graph whose nodes are classes and whose links are is-a\* relationships





# ASSO Revisited Partitioning

Structured Database Schema: designed in order to permit the Revisited Partitioning applicability

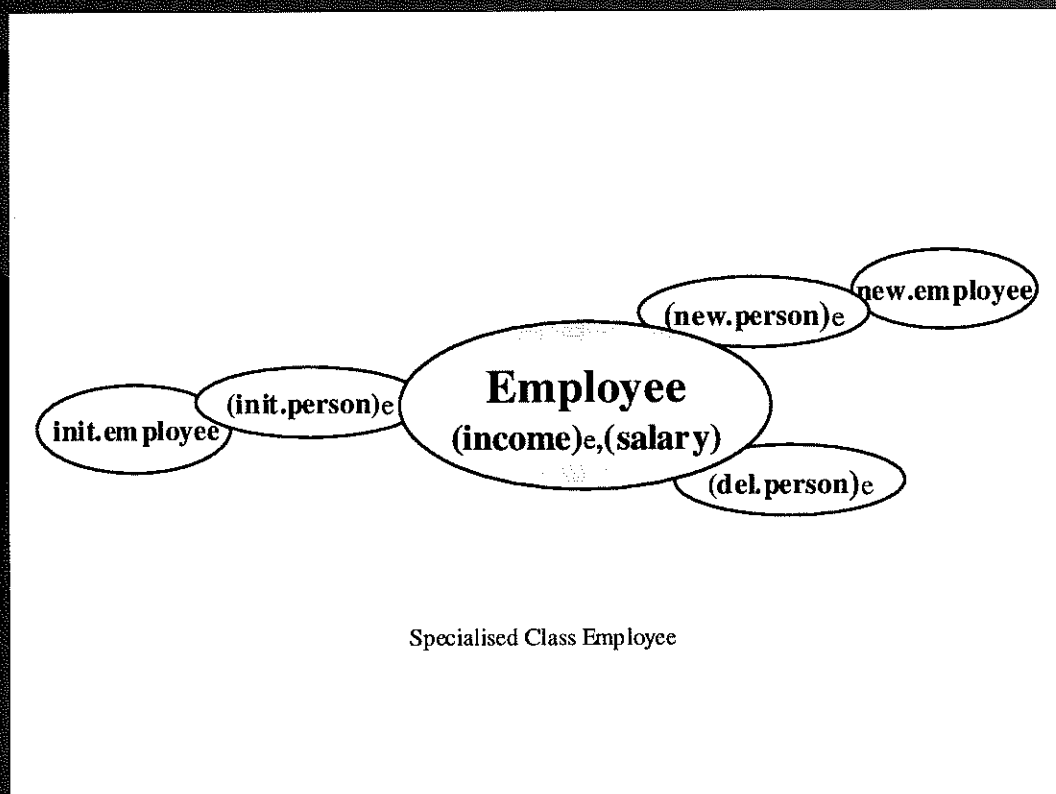


# Specialised Class

## Class with

- attributes and operations inherited from the superclass
- specific operations (specialisations) defined on the class
- possible specific attributes

Specialisations designed in order to preserve the is-a constraints – Consistency to be proved only with respect to application constraints



# ASSO model: second view

**Structured Database Schema:** collection composed by a root class and specialised classes

**Consistency:** the ASSO model is consistent if each class of the collection is consistent

**Proof reduction:** the consistency proof of specialised class can be reduced to the proofs of the corresponding classes

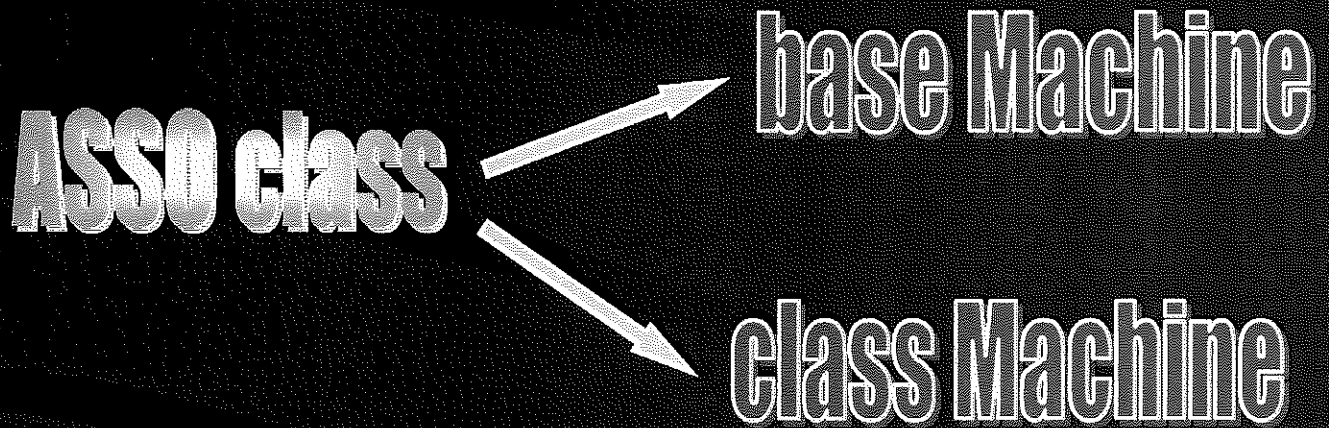
**ASSO and B:** each class is a B-Machine preserving the class constraints



# ASSO - toolkit

Reuse of support tools for B and ASSO features

⇒ Translation Tool



⇒ Revisited Partitioning Tool

