SEMANTIC INTEGRITY
introduction

CONSTRAINTS:

rules guaranteeing semantic integrity

IMPORTANT ASPECTS:

- **SPECIFICATION**
- **VERIFICATION**
- **ENFORCEMENT**

CLASSIFICATION:

- **INHERENT**

  included by the structures of the data model

  (e.g. relatability, convertibility).
- **EXPLICIT**

  additional constraints:

  - **static constraints**
  - **dynamic constraints**
- **IMPLICIT**

  logical consequences of other constraints

INHERENT CONSTRAINTS

- **TYPE SPECIFICATIONS**
  - ***type*** **A1 = B1, B2, ..., Bn**
    type **A2 = B1, B2, ..., Bm**
    are rejected for positive n equaling m.
  - ***type*** **A = B, ..., B**
    is meaningless.
  - ***type*** **A = [X_A], ...**
    is meaningless.
  - ***type*** **A = [X_B], ..., [Y_B]**
    is also meaningless.

- **VALUE RANGES**
  e.g. representation,
  enumeration,
  range,
  pattern.

STATIC CONSTRAINTS

VIRTUAL ATTRIBUTES

EXAMPLE:

type **invoice = customer, date, ...**
type **invoice line = invoice, article, quantity,**
unit_price,...

```
    assert invoice line its amount (0..*) =
        quantity * unit_price.

    assert invoice its invoice_amount (0..*) =
        total invoice line its amount per invoice.

    assert invoice its number of lines (1..*) =
        count invoice line per invoice.
```

DYNAMIC CONSTRAINTS

- **INSERT CONSTRAINT**
  - **INITIALIZATION**

    default **invoice *its* date = ’system date’.**
  - **INSTANTANEOUS RELATIONSHIP**

    init **invoice line *its* unit_price = article *its* price.**

- **DELETE CONSTRAINT**

  (not apart from inherent or static constraints)

- **UPDATE CONSTRAINT**

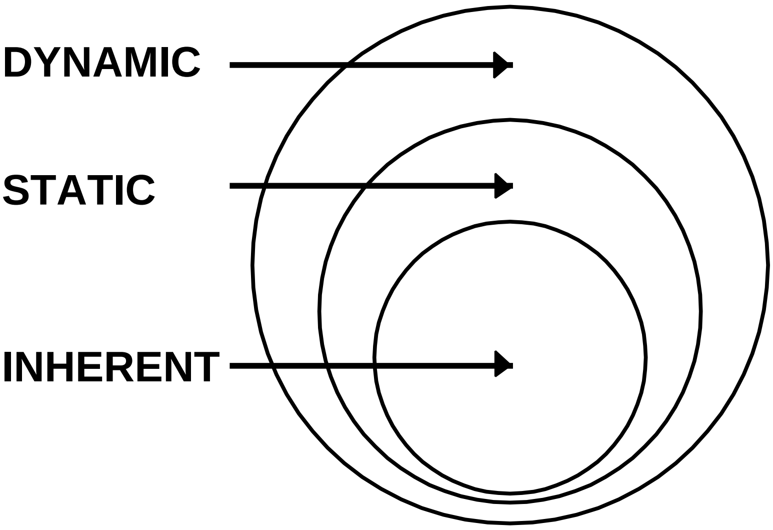
  type **vehicle   = model, serial number, ...**
  type **transfer = vehicle, date, make, ...**
  check **transfer *its* make =**

  case of

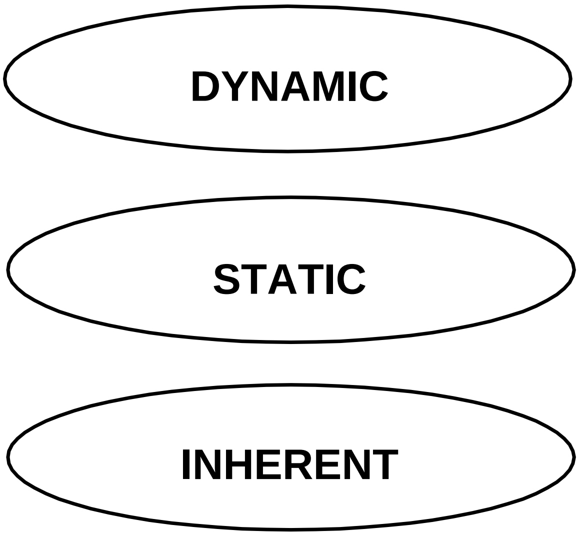
supply	**: purchase,**
purchase	**: purchase,**
purchase	**: exchange,**
exchange	**: purchase,**
exchange	**: scrap,**
default	**: supply.**

CONCLUSION
SPECIFICATION AND VERIFICATION

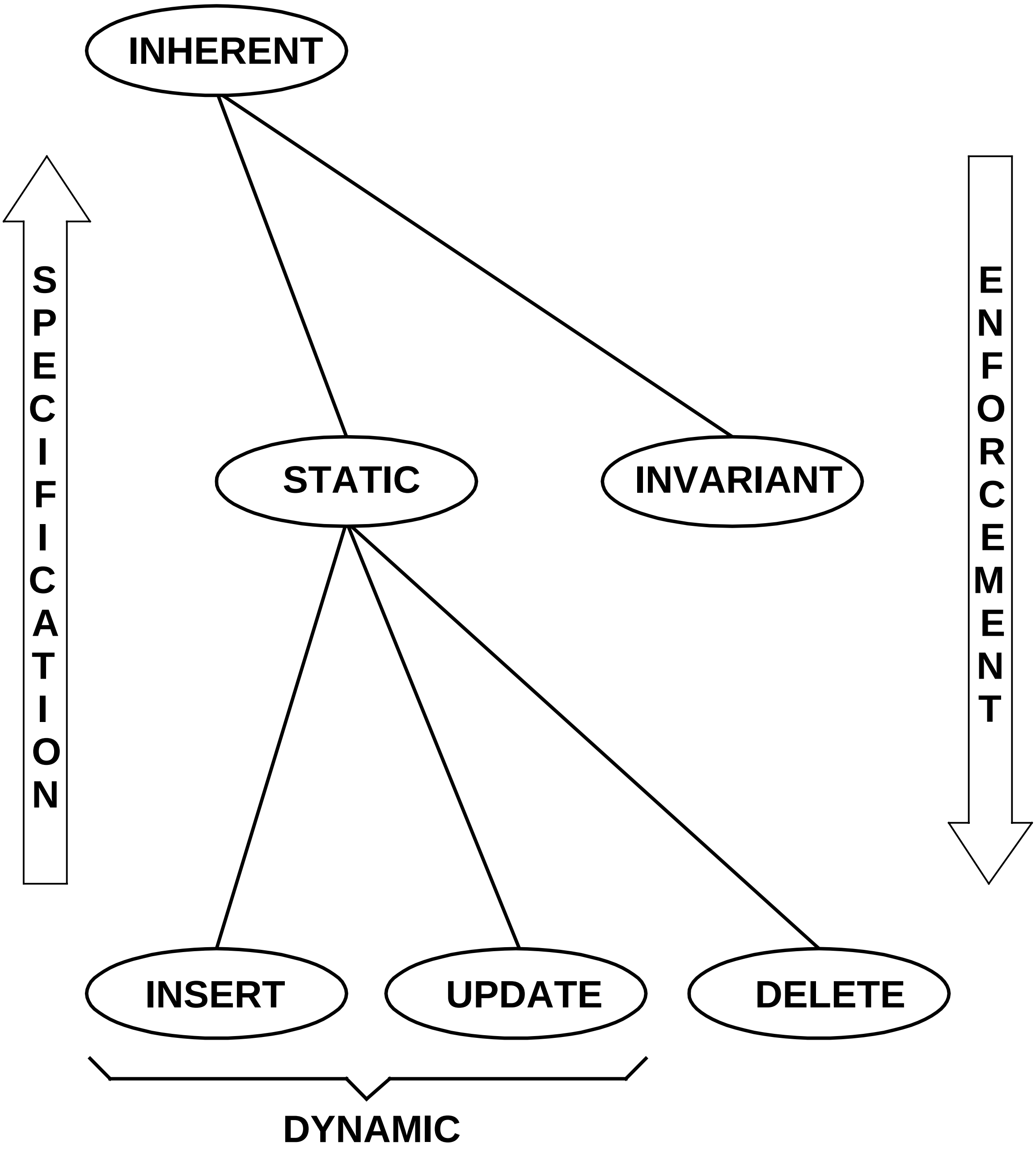
- **IN GENERAL: OVERLAPPING CATEGORIES.**
  PROBLEM: EXTREMELY DIFFICULT TO VERIFY.



- **IN XPLAIN: DISJOINT CATEGORIES.**
  ADVANTAGE: EASIER TO VERIFY.



CONCLUSION
SPECIFICATION AND ENFORCEMENT



EXERCISES

EXERCISE 1.

The relationship between employees and their managers is given in the following type definition:

***type* employee = name, salary, ..., manager_employee.**

Provide the static constraint enforcing managers to have higher salaries than their employees.

EXERCISE 2.

Provide a type definition describing the relationship between persons and their parents. Also describe the person’s date of birth. Formulate the static constraint enforcing each person to be younger than the parents.

EXERCISES
continued

EXERCISE 3.

The following type definitions are available:

type **department = address, extension**
type **employee = name, extension, department.**

Employees can be reached by telephone on the department extension after employment. Specify the dynamic constraint.

Can telephone numbers be changed?

Indicate possible consequences of changes for employees and departments.