Computer Science Program, The University of Texas, Dallas

From SADT
to a Formal OO-RML

SADT Revisited

- SADT: ambiguities persist in narrative
  - ***Boxes inside a box may represent***

    specializations: e.g., A1 isA A0

    instances: e.g., A1 instanceOf A0
    aggregation: e.g., A1 partOf A0

    of the concept represented by the box

  - ***Temporal relationships are not clear:***

    When are inputs produced?
    as a chunk, in a piece-meal, upon request
    When are outputs produced?
    immediately after receiving inputs, anytime, according to controls
    When do boxes perform actions?
    sequentially, concurrently throughout, partially overlapping

  - ***Attributes of data not easy to express:***

    How do we aggregate {name, age, address, ...} of a person?

  - ***Constraints on arrows now easy to express:***

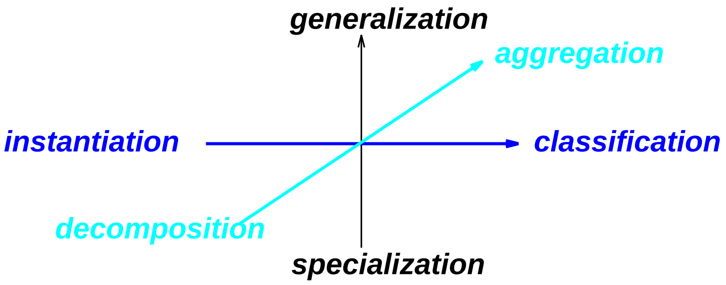
    Are some inputs optional/mandatory?
    Are any inputs legal?
    What is the difference between inputs and outputs?
    Are controls always clearly different from mechanisms?

Primitives of OO-RML

* Ontological primitives:

entities: objects in the domain of discourse
activities: induce changes in the world
assertions: constraints on the behavior of entities and activities

* epistemological (abstraction/structural) primitives:



generalization

aggregation

instantiation

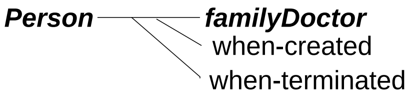
classification

decomposition

specialization

v ***aggregation/decomposition:*** ***categories of attributes***
relationships between entities/objects are treated also as objects

Example1:



Person

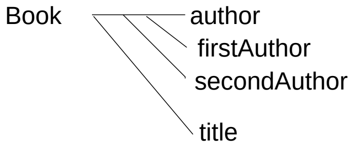
familyDoctor
when-created
when-terminated

EntityClass Person
familyDoctor: MD
when-created: Date
when-terminated: Date

Primitives of OO-RML

v aggregation/decomposition: categories of attributes
relationships between entities/objects are treated also as objects

Example2:



Book

author
firstAuthor
secondAuthor
title

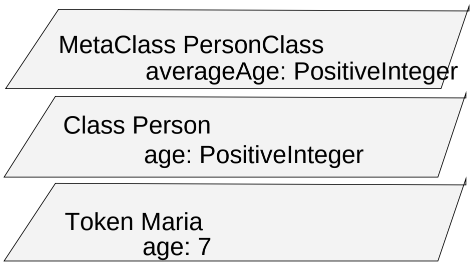
EntityClass Book
author
firstAuthor: Person
secondAuthor: Person
title: string

v generalization/specialization:
infinite hierarchy of classes
tokens, classes, metaclasses, metametaclasses, ... omegaclass

Recall:

an object is a "concept" of anything (inclusing concept)
each class is a concept
each class is an object
each object belongs to a class

each class belongs to some class!!



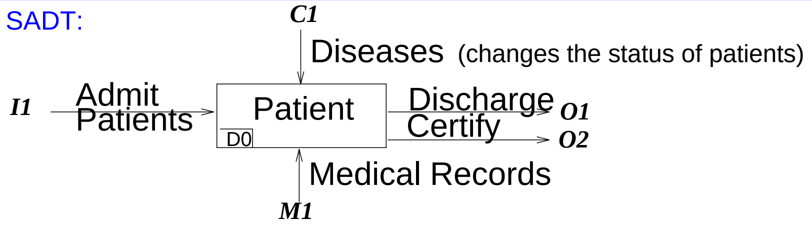
MetaClass PersonClass
averageAge: PositiveInteger

Class Person
age: PositiveInteger

Token Maria
age: 7

Entities

* SADT:



* OO-RML

EntityClass Persons with
  necessary part
    name: Names
    gender: {'Male, 'Female}
  association
    address: Addresses
    nearestRelative: Persons
    familyDoctor: MD
      when-created: Date
      when-terminated: Date
    favoriteBook: Book

EntityClass Patients with
  necessary unique part
    record: Medical-Records
  association
    location: Nursing-Homes
    room: Rooms
    Physician: MD
    diagnosis: Diseases
    paymentDue: $Values
  producer
    register: AdmitPatients (person=this, toHome=loc)
  consumer
    release: Discharge (patient=this)
    decease: Certify (certifee=this, status=dead)
  initially
    startClean?: (paymentDue=0)

; *part: properties which express relationships which ordinarily do not change with time*

; *association: the property value may change over time*

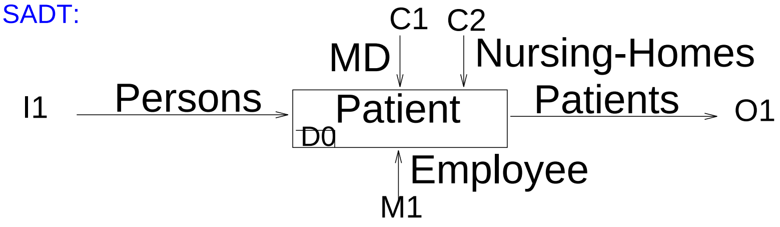
; *necessary: attribute each instance should have always*

; *producer: the property value is an event, one of whose effects is to make a new instance of the class being defined*

; *consumer: the property value is an event one of whose effects is to make an instance of the class being defined stop being an instance of the class*

Activities

* SADT:



* OO-RML

ActivityClass AdmitPatients with
  input
    person:Persons
  control
    toHome: Nursing-Homes
    doc: MD
  mechanism
    cleark: Employee
  output
    patient: Patients
  initially
    already-in?: not (person in Patients)
  finally
    admitted?: (person=patient) and (patient.location=toHome)
  part
    getBasicInfo: Interview (whom=person)
    place: AssignRoom (toWhom=person)
    getConsult: ScheduleVisit (visitor=doc, visitee=patient)
    assess: TakeVitalSigns (visitee=patient)
  integrityConstraint

; activation condition, termination condition
when should an activity start and end

; initially (preconditions) and finally (postconditions):
what conditions should hold for an activity to start;
what conditions should hold at the time of termination