

From SADT to a Formal OO-RML

Lawrence Chung

SADT Revisited

☞ SADT: ambiguities persist in narrative

✧ **Boxes inside a box may represent**

specializations: e.g., A1 isA AO

instances: e.g., A1 instanceOf AO

aggregation: e.g., A1 partOf AO

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?

Lawrence Chung

Primitives of OO-RML

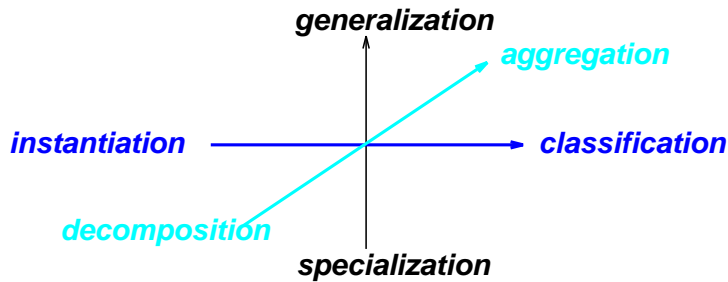
* Ontological primitives:

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:



V **aggregation/decomposition: categories of attributes**

relationships between entities/objects are treated also as objects

Example1:



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

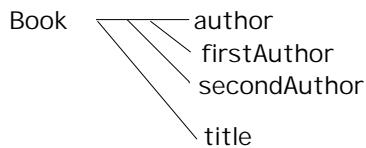
Lawrence Chung

Primitives of OO-RML

V **aggregation/decomposition: categories of attributes**

relationships between entities/objects are treated also as objects

Example2:



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

V **generalization/specialization:**

infinite hierarchy of classes

tokens, classes, metaclasses, metametaclasses, ... omegaclass

Recall:

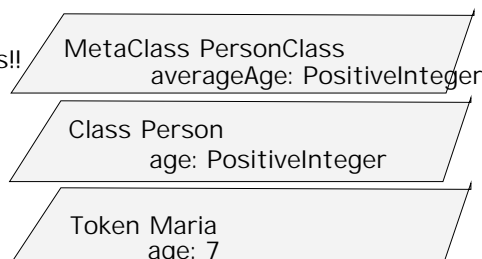
an object is a "concept" of anything (including concept)

each class is a concept

each class is an object

each object belongs to a class

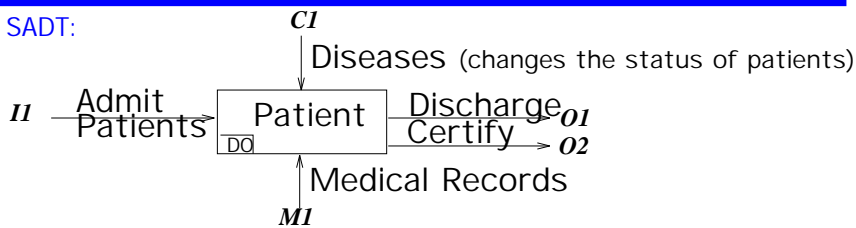
each class belongs to some class!!



Lawrence Chung

Entities

* SADT:



* OO-RML

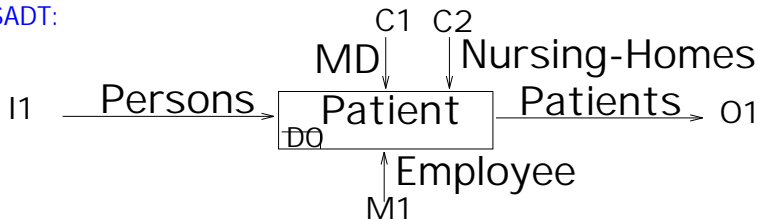
EntityClass Persons with necessary part name: Names gender: {'Male', 'Female'} association address: Addresses nearestRelative: Persons familyDoctor: MD when-created: Date when-terminated: Date favoriteBook: Book	EntityClass Patients with necessary unique part record: Medical-Records association location: Nursing-Homes room: Rooms Physician: MD diagnosis: Diseases paymentDue: \$Values producer register: AdmitPatients (person=this, toHome=loc) consumer release: Discharge (patient=this) decease: Certify (certiffee=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*

Lawrence Chung

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	; <i>activation condition, termination condition when should an activity start and end</i> ; <i>initially (preconditions) and finally (postconditions) what conditions should hold for an activity to start; what conditions should hold at the time of termination</i>
--	--

Lawrence Chung