

Deep Domain Object +

Sebastian Gruber, Bernd Neumayr

30.06.2022

Introduction

Just-in-Time Adaptive Interventions (JITAIs)

- **JITAI** is an intervention design to support health behavior changes
 - "provide the right type/amount of support",
 - "at the right time",
 - "by adapting to an individual's changing internal and contextual state"

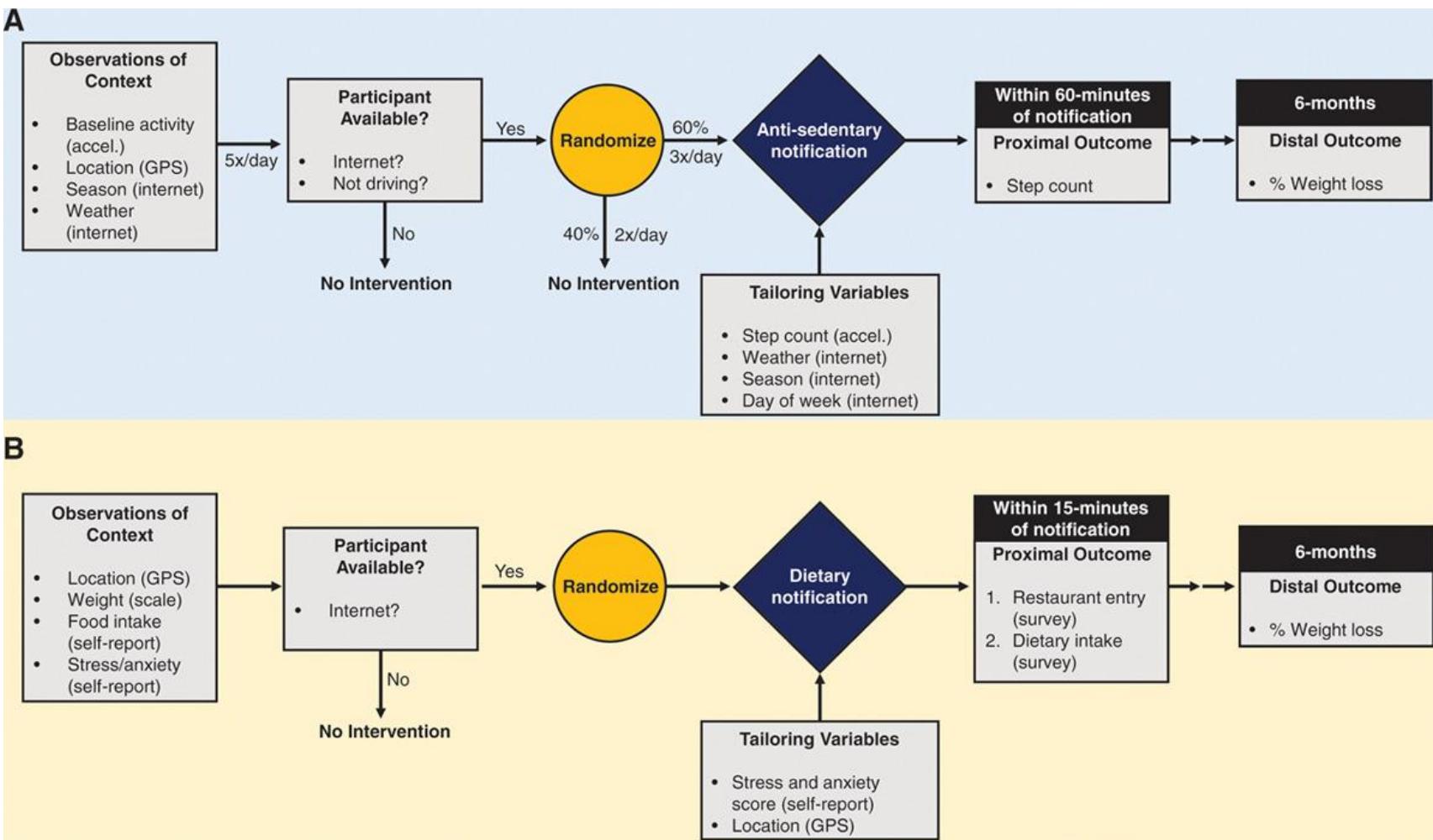
(Nahum-Shani et al., 2018)

- Wide range of applications
 - Physical inactivity
 - Alcohol use
 - Mental illness
 - Smoking
 - Obesity

(Nahum-Shani et al., 2018)

Jane is available for intervention and has been sedentary for the last 60 minutes. The system suggests to take a break from sitting and do 2 minutes of air squats.

Example



Multi-level Modeling for Customization of JITAls

- Goal: **integration-preserving customization**
 - Facilitate customization (heterogeneity)
 - e.g., customization of decision rules, machine learning, ...
 - Preserve data integration (homogeneity) for data analytics
 - e.g., machine learning over JITAI outcomes of different participants
 - e.g., data aggregation over JITAI outcomes in different JITAI studies
- Solution Idea: a kind of **multi-level object-oriented modeling**
 - Derive extensive hierarchies of compositions, sub classes and meta classes
 - Use fine-grained sub class hierarchies for customization
 - Use meta class hierarchies for data aggregation and meta schema constraints

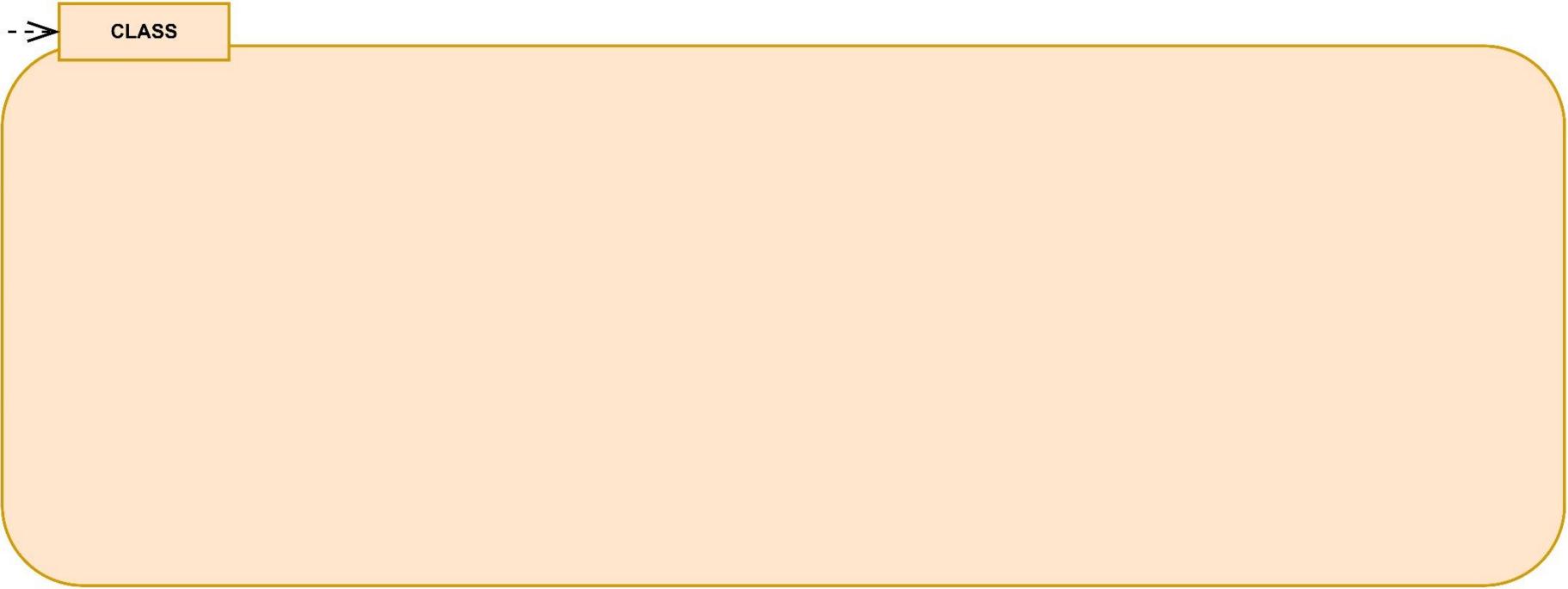
Core

We create a domain model for the
Ludwig Boltzmann Institute for
Digital Health and Prevention (LBI)
which is a research organization
represented by the URI
`<http://dhp.lbg.ac.at>`.

```
<http://dhp.lbg.ac.at>
    rdf:type ddo:DomainObject .
```

<http://dhp.lbg.ac.at>

CLASS



<http://dhp.lbg.ac.at>

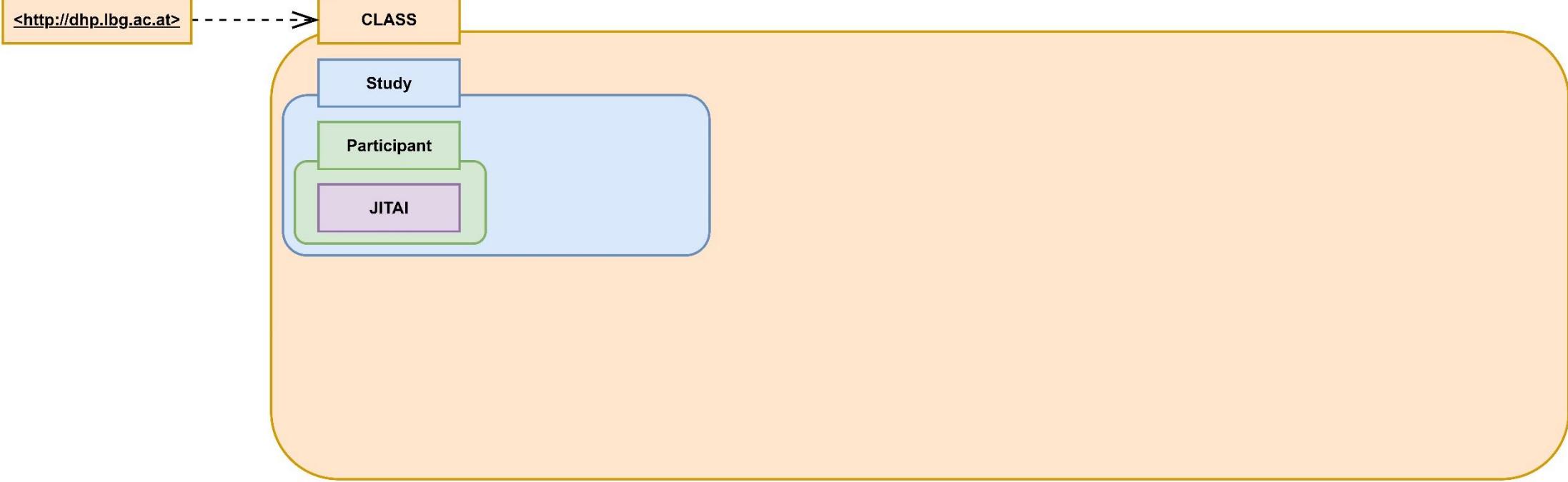
CLASS

The LBI conducts studies for developing and evaluating the effectiveness of JITAIs on study participants.

```
<CLASS/Study>
    rdf:type ddo:DomainObjectClass , ddo:ModeledClass ;
    ddo:name "Study" ;
    ddo:parent <CLASS> .

<CLASS/Study/Participant>
    rdf:type ddo:DomainObjectClass , ddo:ModeledClass ;
    ddo:name "Participant" ;
    ddo:parent <CLASS/Study> .

<CLASS/Study/Participant/JITAI>
    rdf:type ddo:OccurrenceClass , ddo:ModeledClass ;
    ddo:name "JITAI" ;
    ddo:parent <CLASS/Study/Participant> ;
    ddo:leafs ddo:AbstractClass .
```



<http://dhp.lbg.ac.at>

CLASS

Study

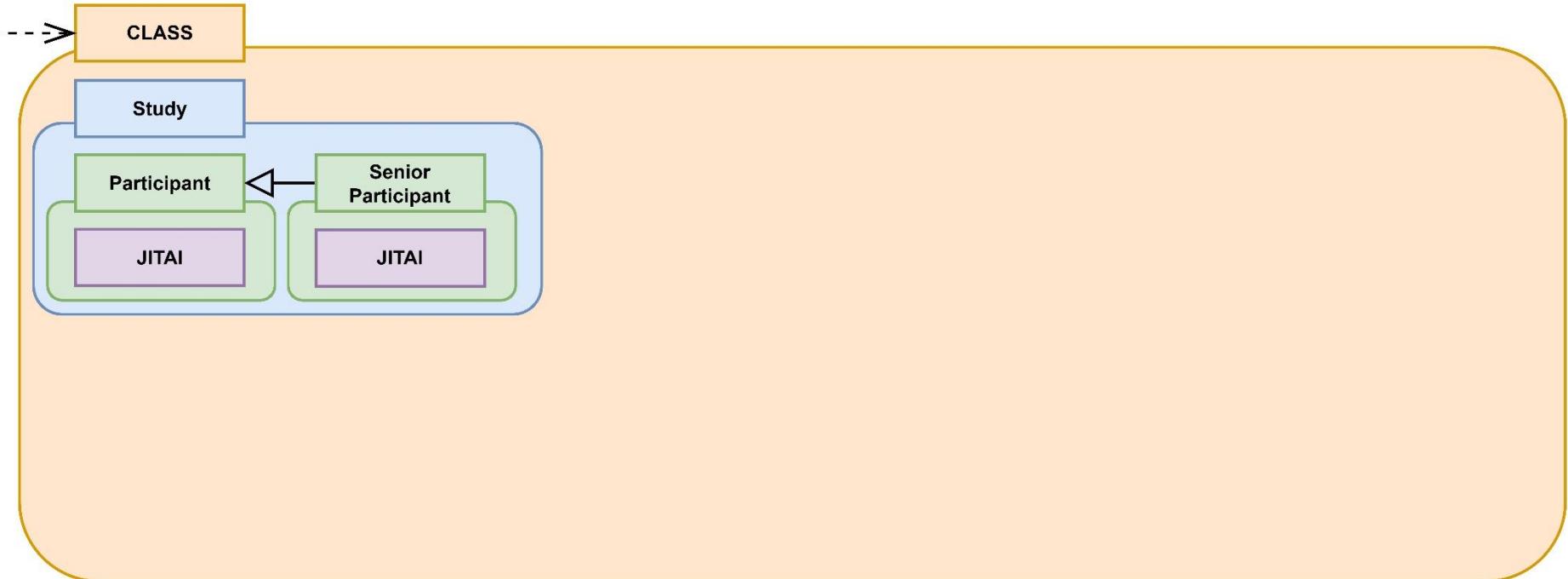
Participant

JITAI

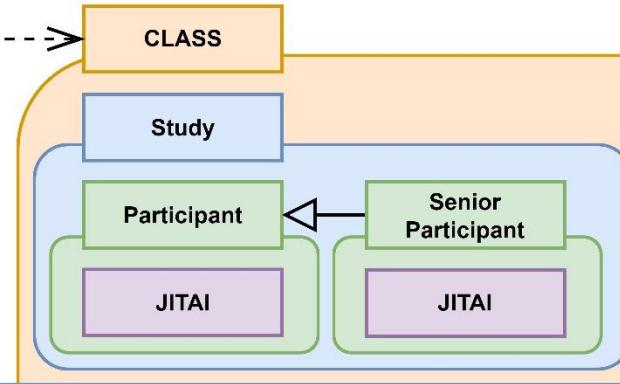
Participants who are considered senior citizens (65 years and older) require special customization.

```
<CLASS/Study/SeniorParticipant>
    rdf:type ddo:DomainObjectClass , ddo:ModeledClass ;
    ddo:name "SeniorParticipant" ;
    ddo:parent <CLASS/Study> ;
    ddo:specializationOf n:Participant .
```

<http://dhp.lbg.ac.at>

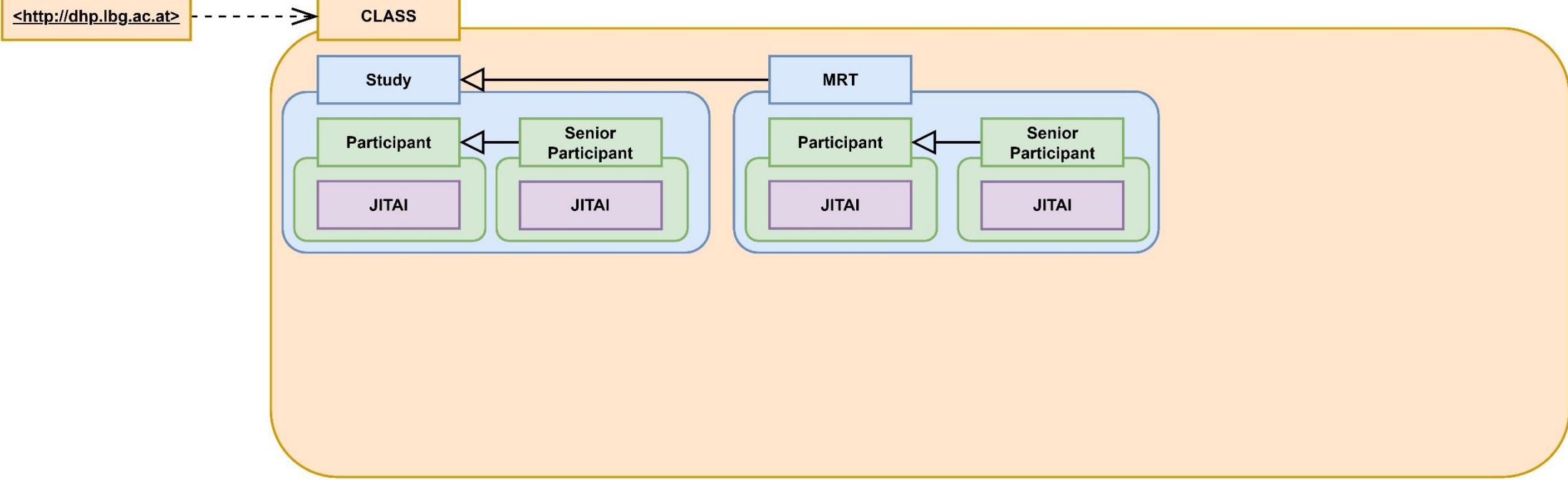


<http://dhp.lbg.ac.at>

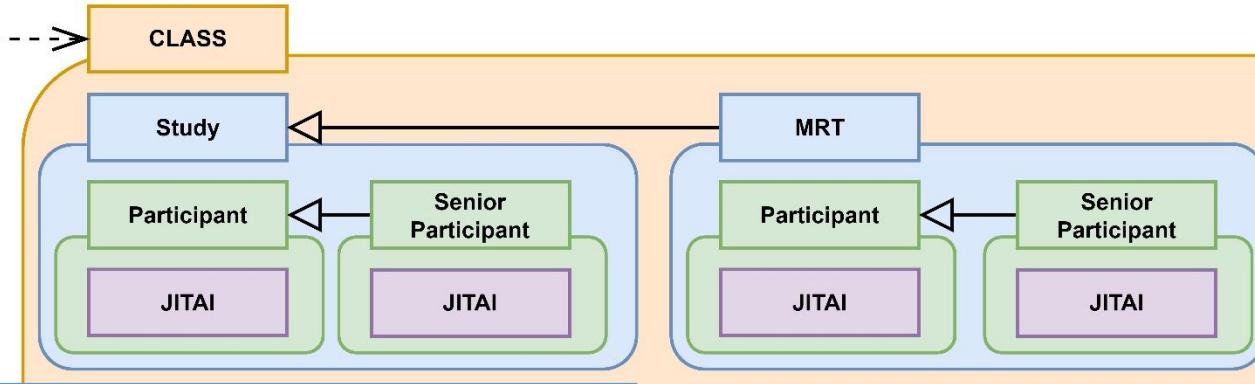


The micro-randomized trial (MRT) is a study design for optimizing JITAIs by sequentially randomizing every intervention decision (within-subject).

```
<CLASS/MRT>
    rdf:type ddo:DomainObjectClass , ddo:ModeledClass ;
    ddo:name "MRT" ;
    ddo:parent <CLASS> ;
    ddo:specializationOf n:Study .
```



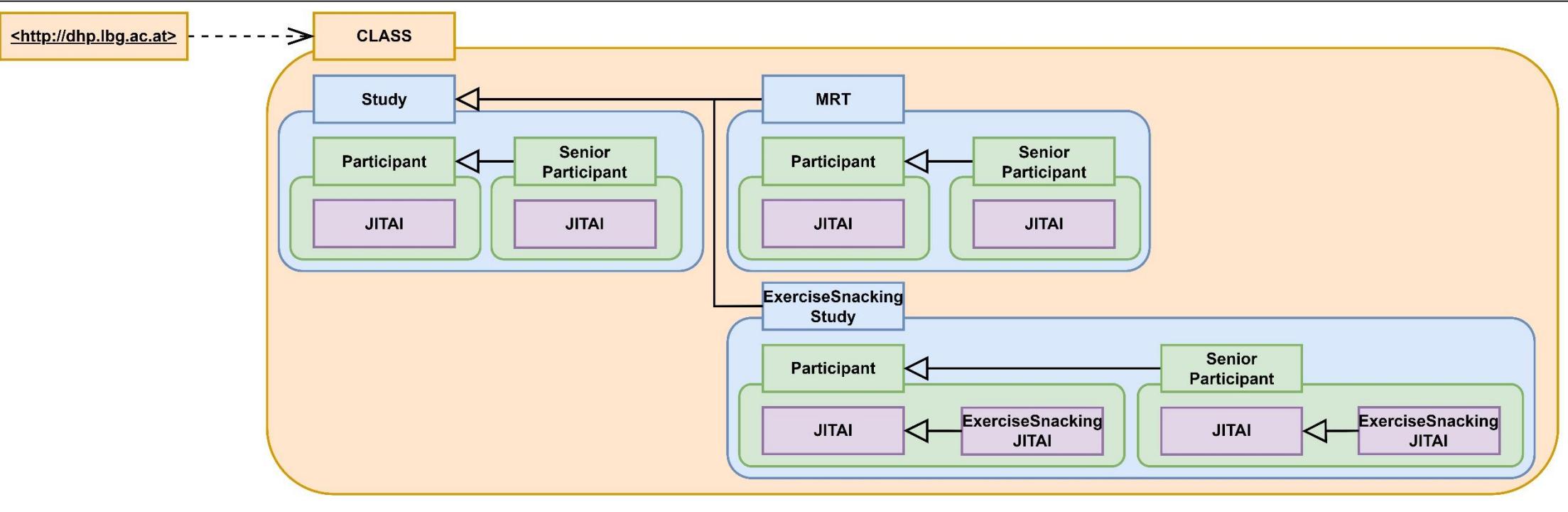
<http://dhp.lbg.ac.at>



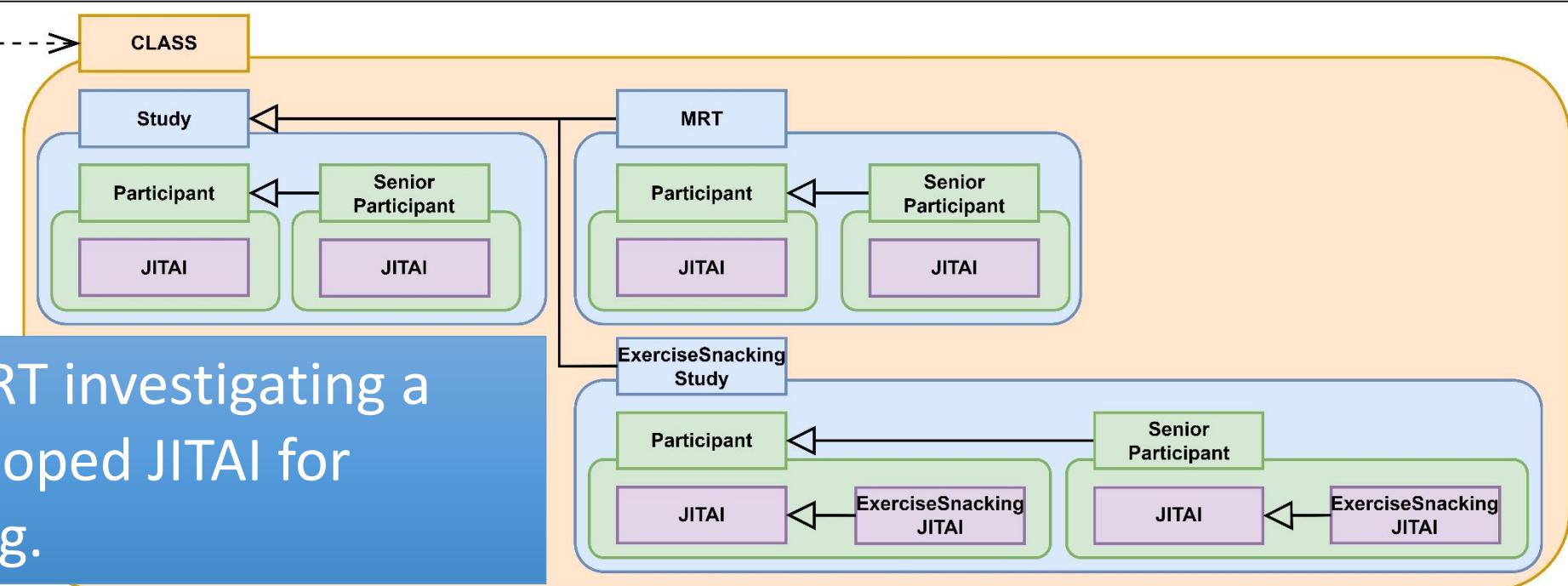
The LBI wants to develop and investigate a JITAI for exercise snacking in multiple studies.

```
<CLASS/ExerciseSnackingStudy>
    rdf:type ddo:DomainObjectClass , ddo:ModeledClass ;
    ddo:name "ExerciseSnackingStudy" ;
    ddo:parent <CLASS> ;
    ddo:specializationOf n:Study .

<CLASS/ExerciseSnackingJITAI>
    rdf:type ddo:OccurrenceClass , ddo:ModeledClass ;
    ddo:name "ExerciseSnackingJITAI" ;
    ddo:parent <CLASS/ExerciseSnackingStudy/Participant> ;
    ddo:specializationOf n:JITAI ;
    ddo:leafs ddo:ConcreteClass .
```

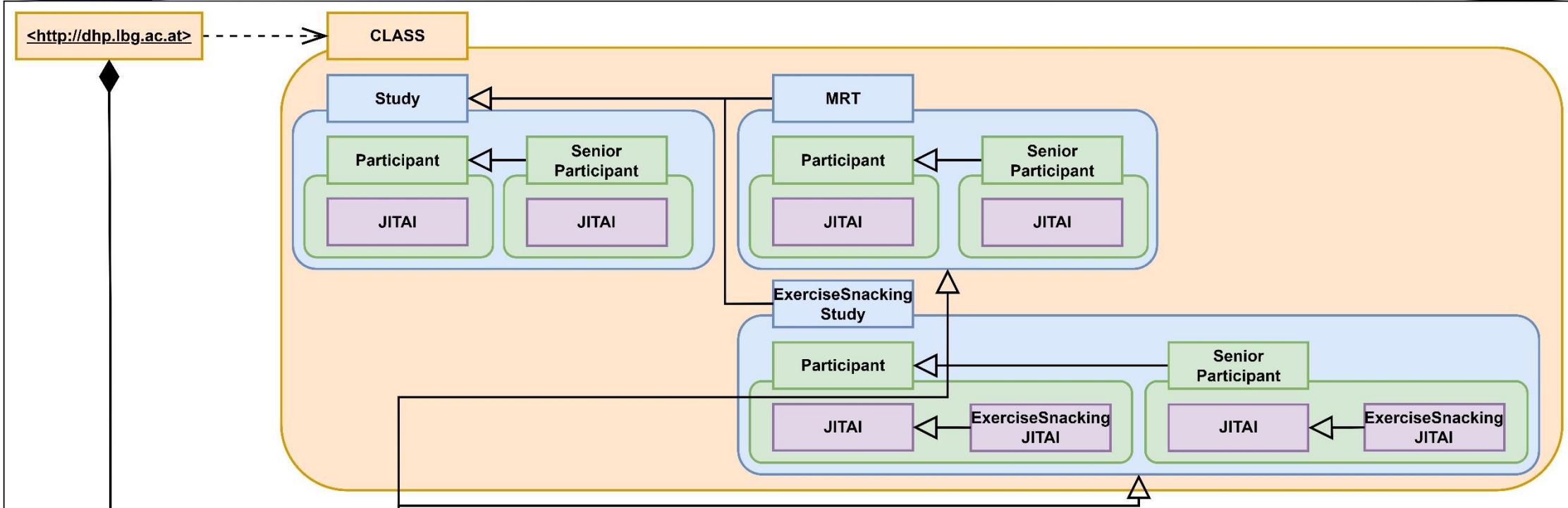


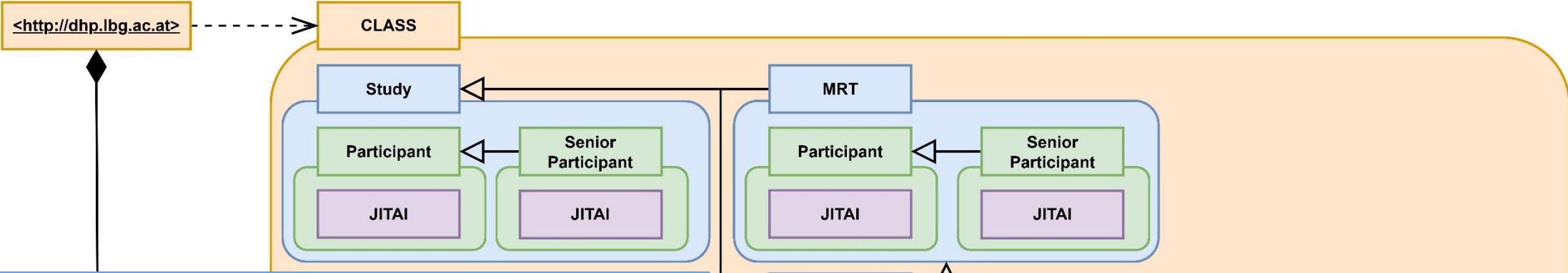
<http://dhp.lbg.ac.at>



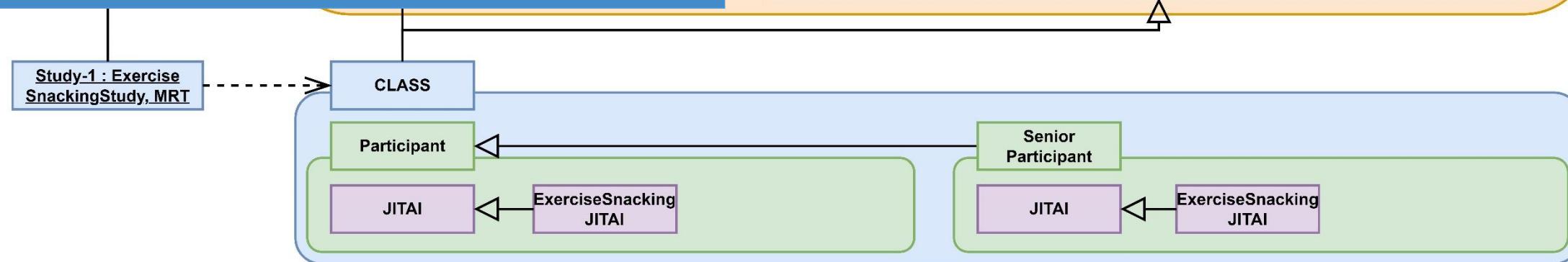
Study-1 is an MRT investigating a previously developed JITAI for exercise snacking.

```
<Study-1>
    rdf:type ddo:DomainObject ;
    ddo:name "Study-1" ;
    ddo:parent <http://dhp.lbg.ac.at> ;
    ddo:instanceOf n:MRT , n:ExerciseSnackingStudy .
```

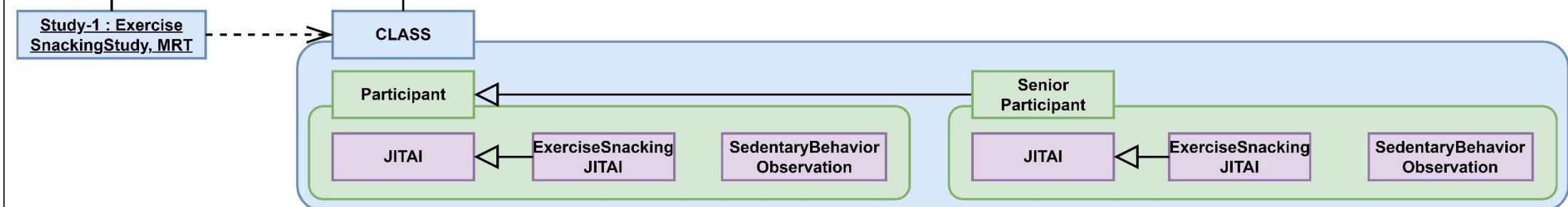
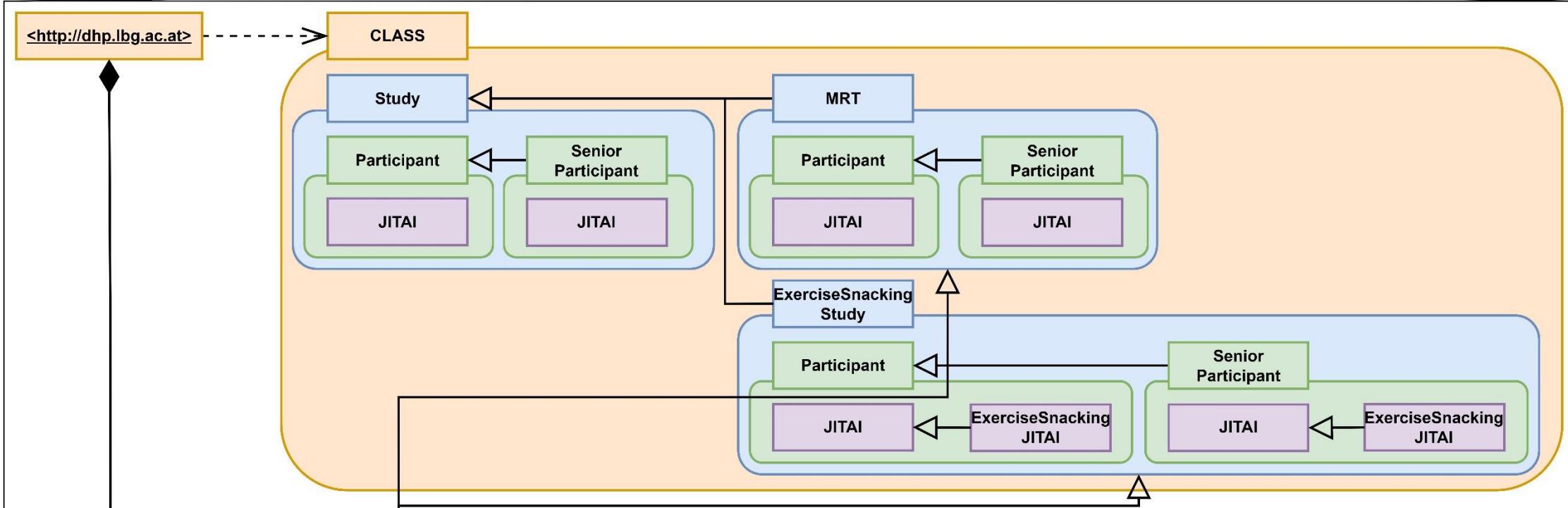


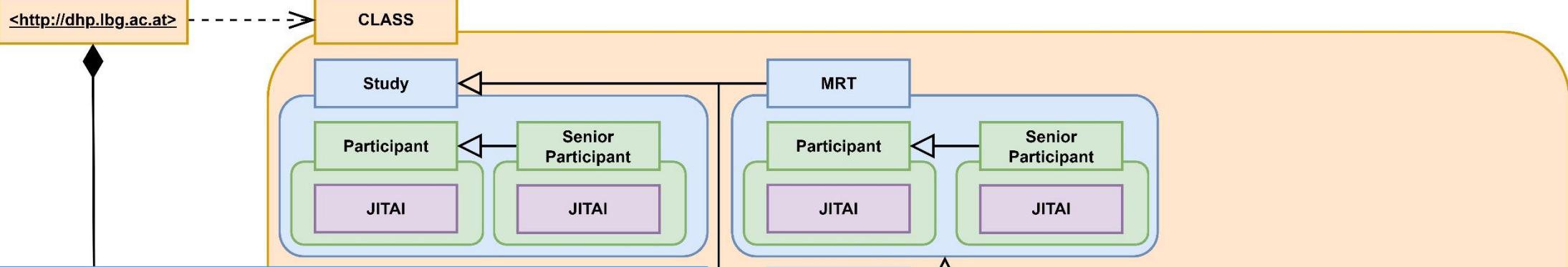


In Study-1, the sedentary behavior of participants is observed for making intervention decisions.

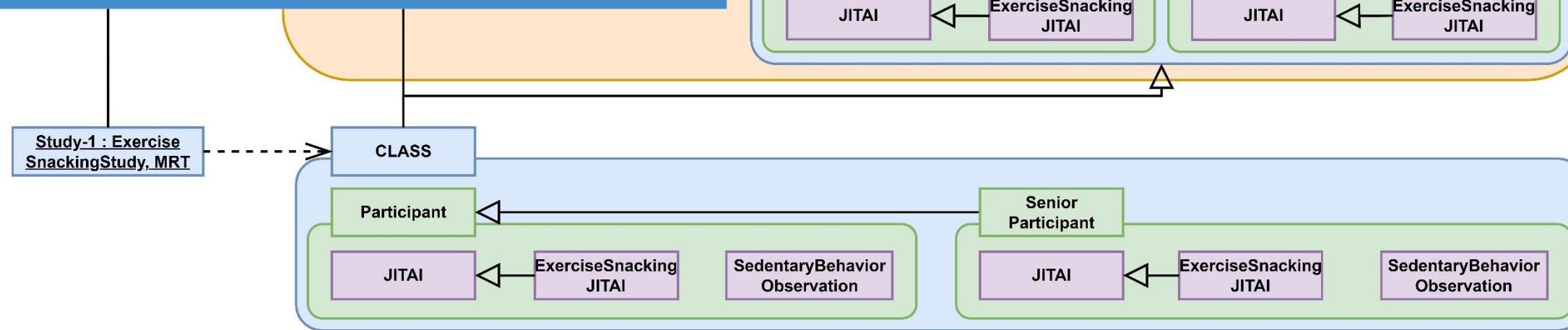


```
<Study-1/CLASS/Participant/SedentaryBehaviorObservation>
  rdf:type ddo:OccurrenceClass , ddo:ModeledClass ;
  ddo:name "SedentaryBehaviorObservation" ;
  ddo:parent <Study-1/CLASS/Participant> ;
  ddo:leafs ddo:ConcreteClass .
```





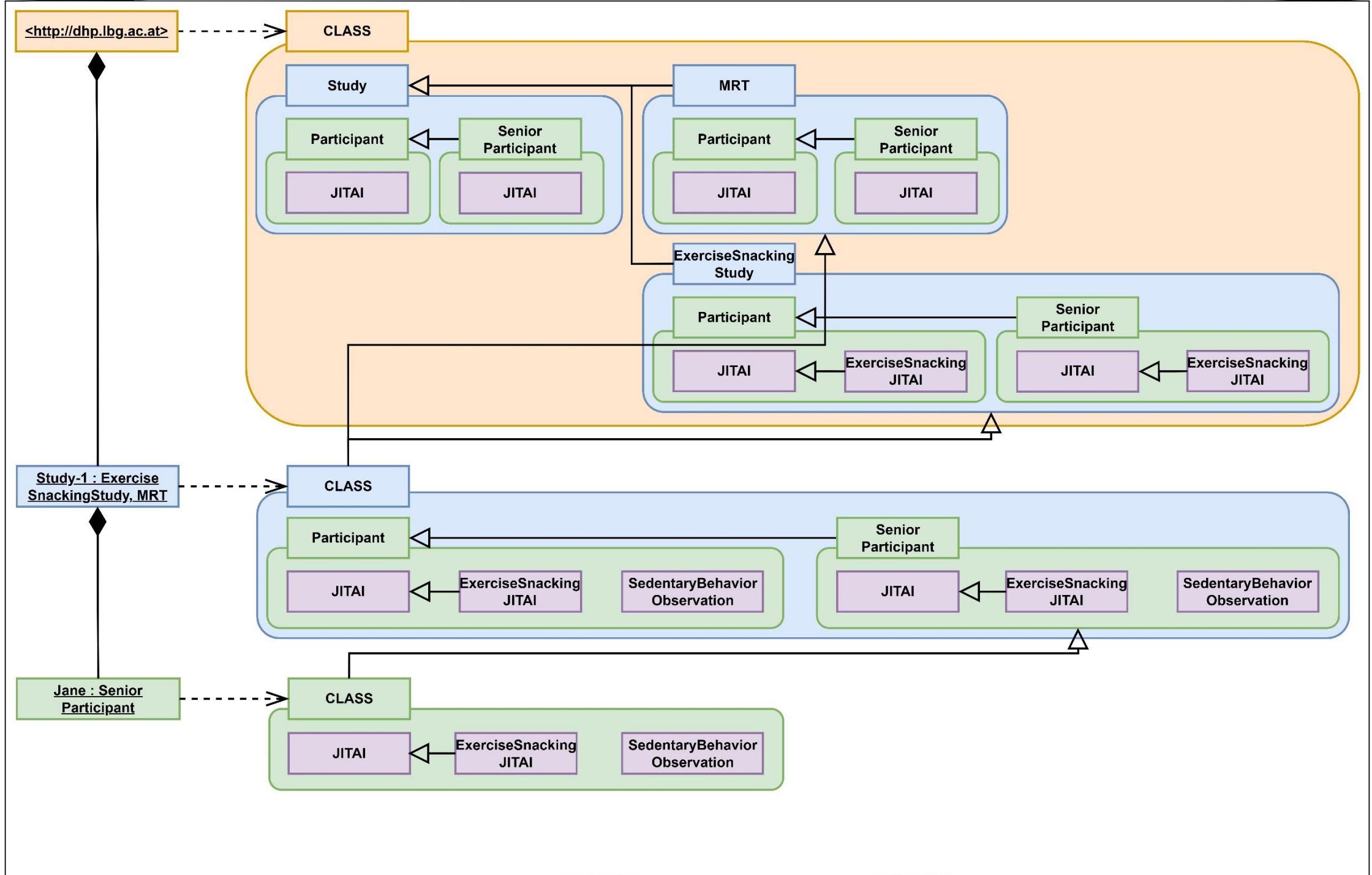
Jane, a senior citizen, takes part in Study-1.



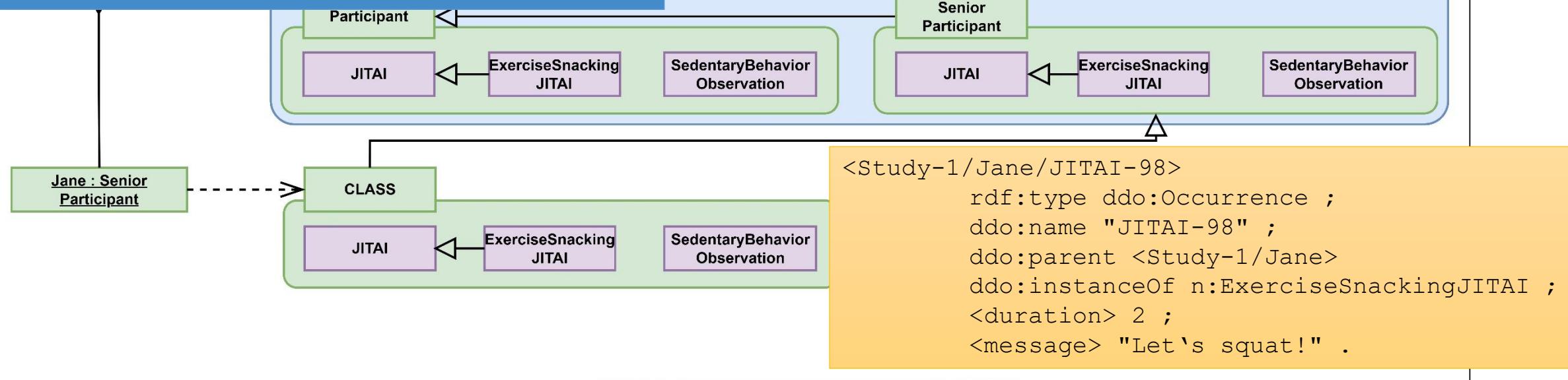
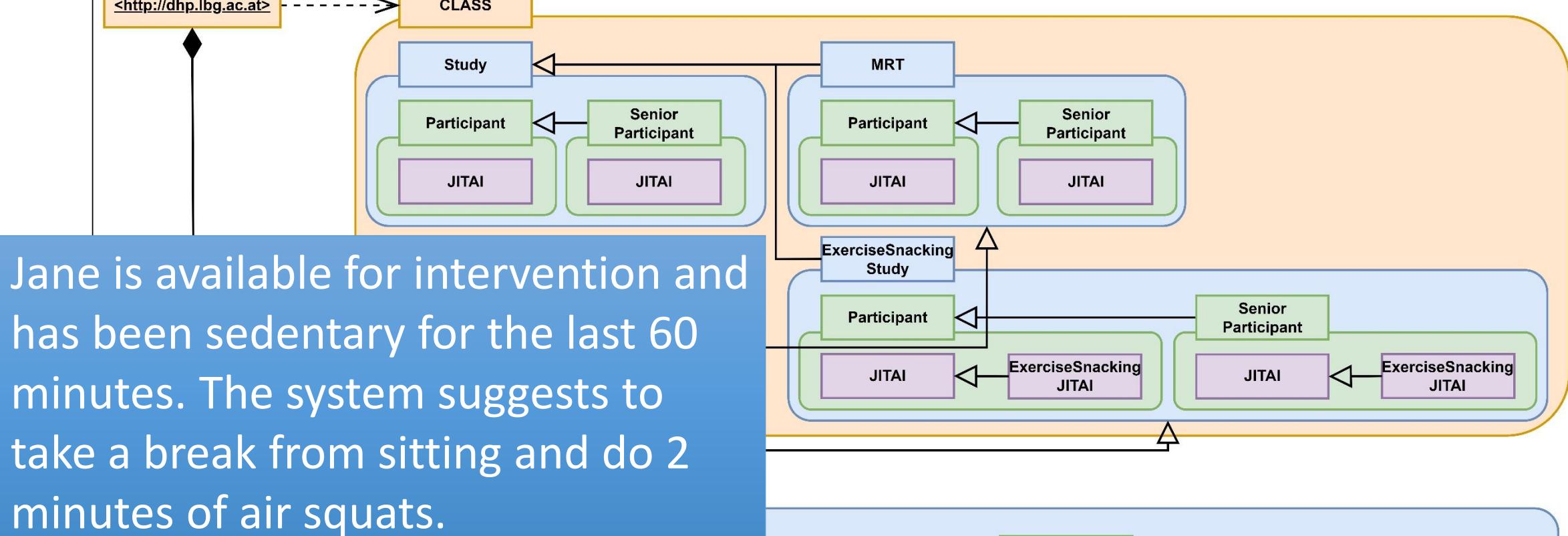
<Study-1/Jane>

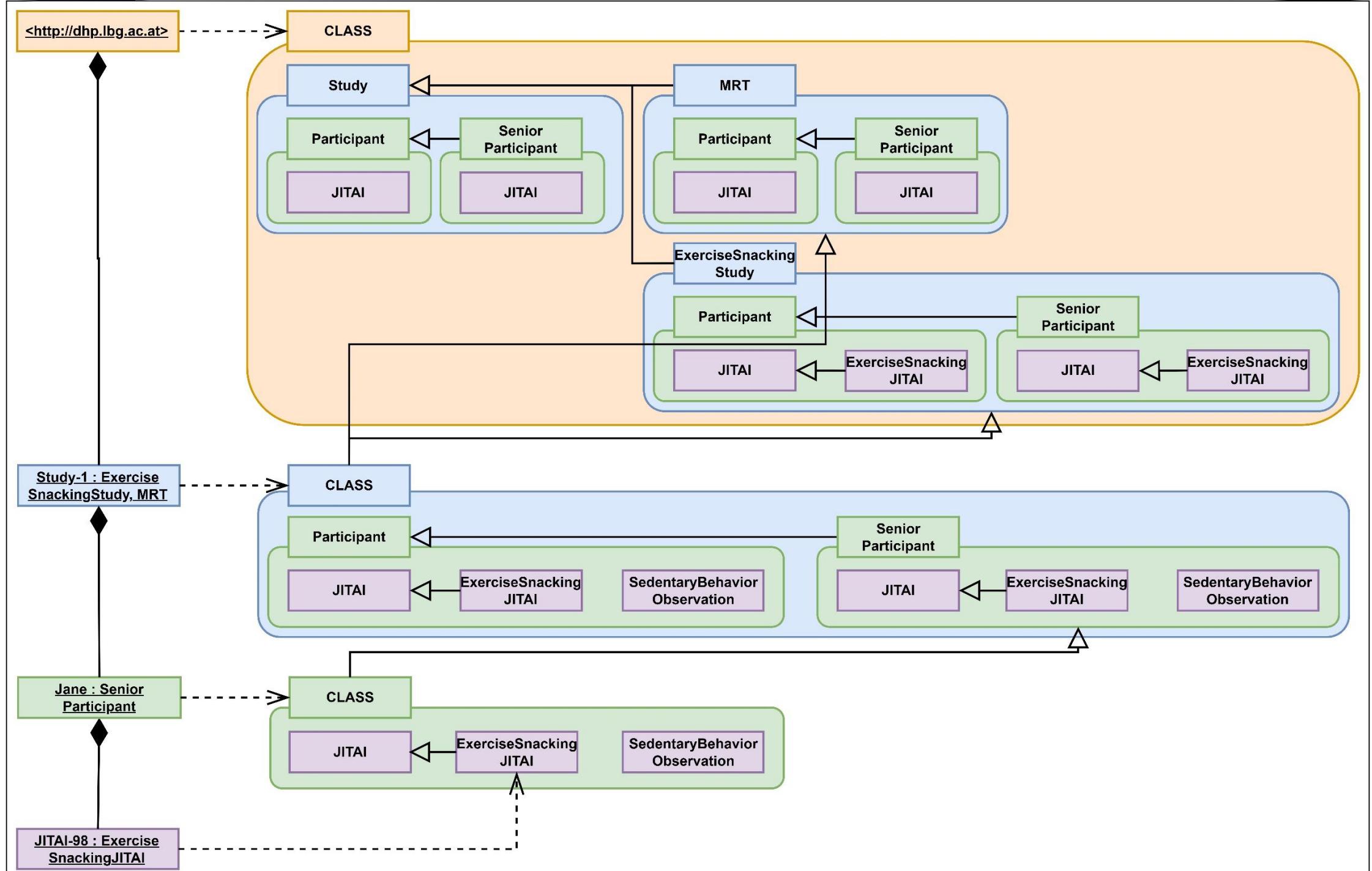
```

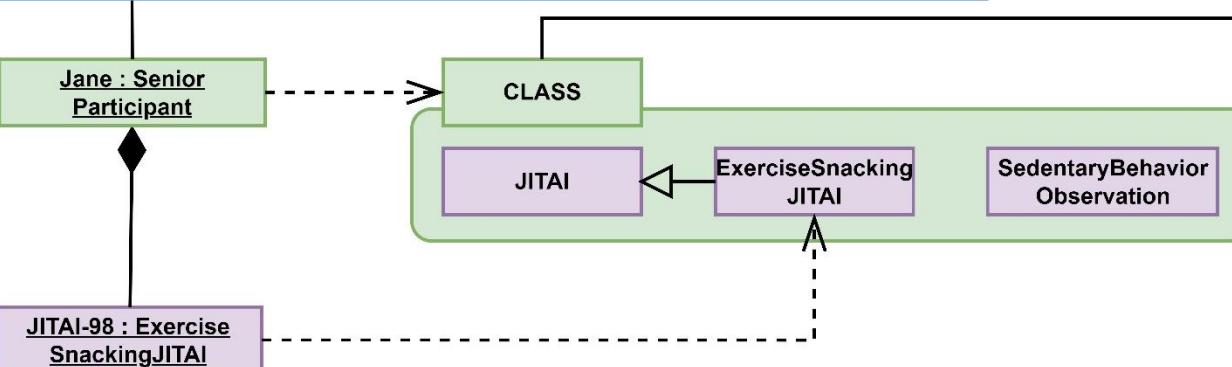
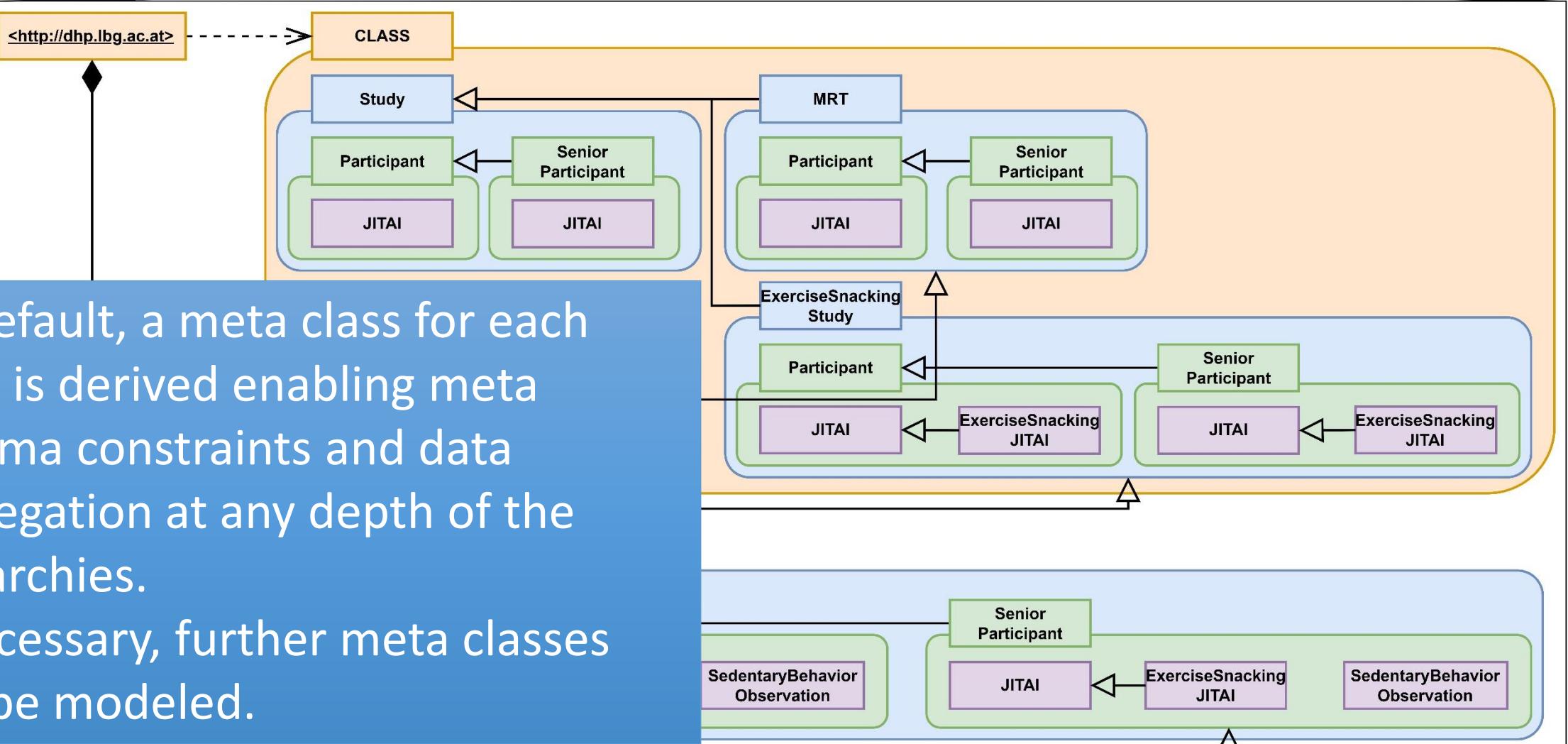
rdf:type ddo:DomainObject ;
ddo:name "Jane" ;
ddo:parent <Study-1> ;
ddo:instanceOf n:SeniorParticipant .
  
```

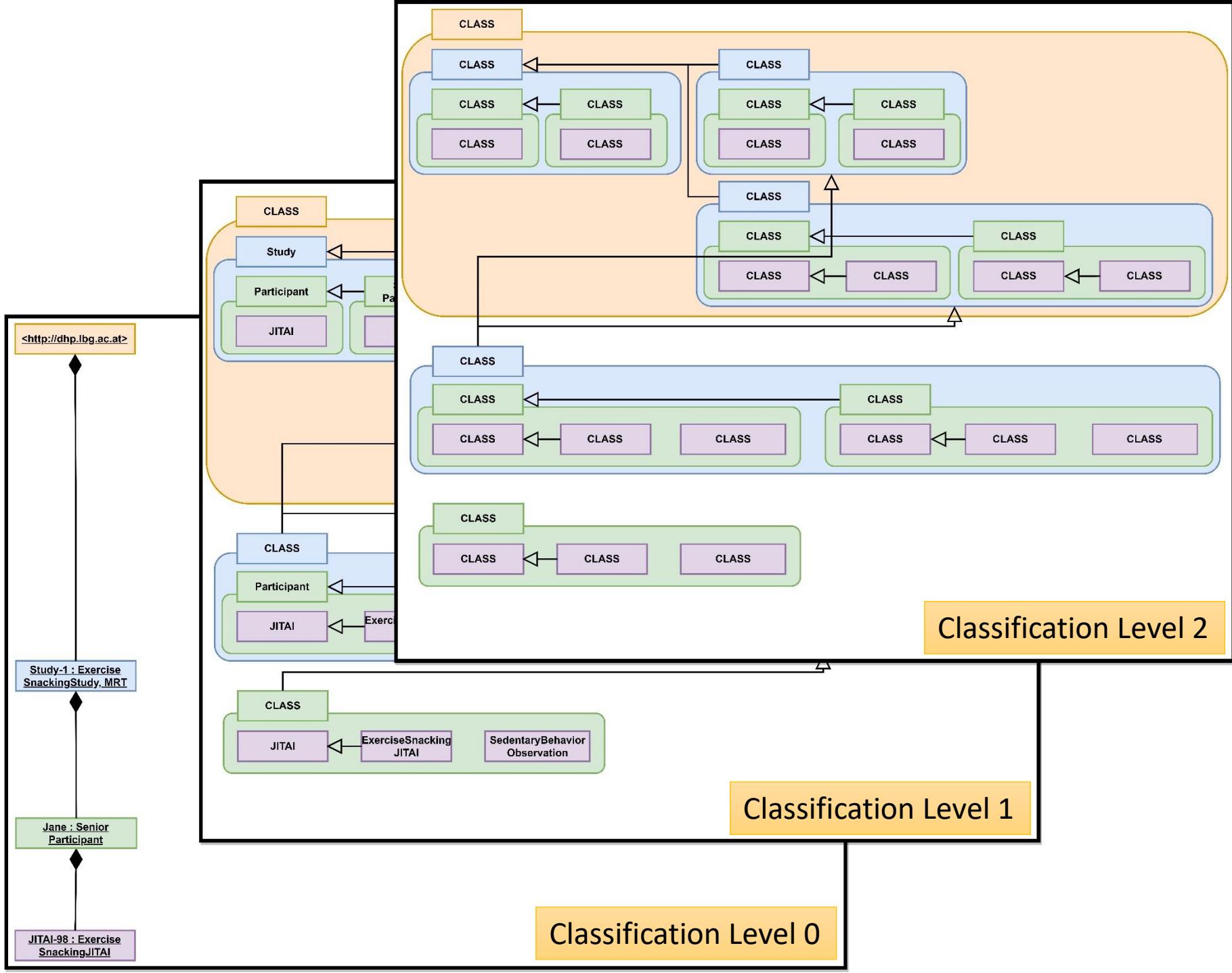


<http://dhp.lbg.ac.at>

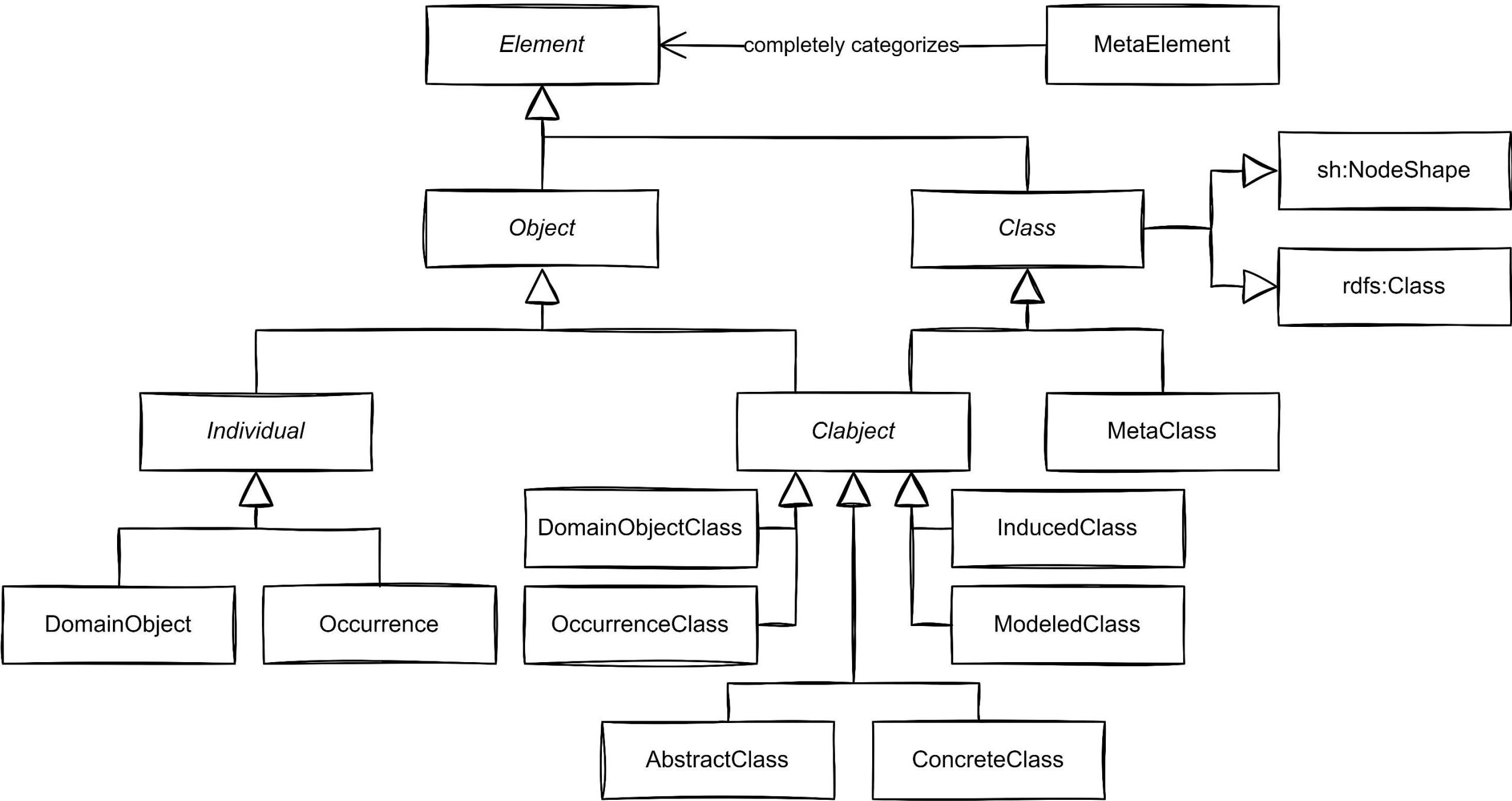




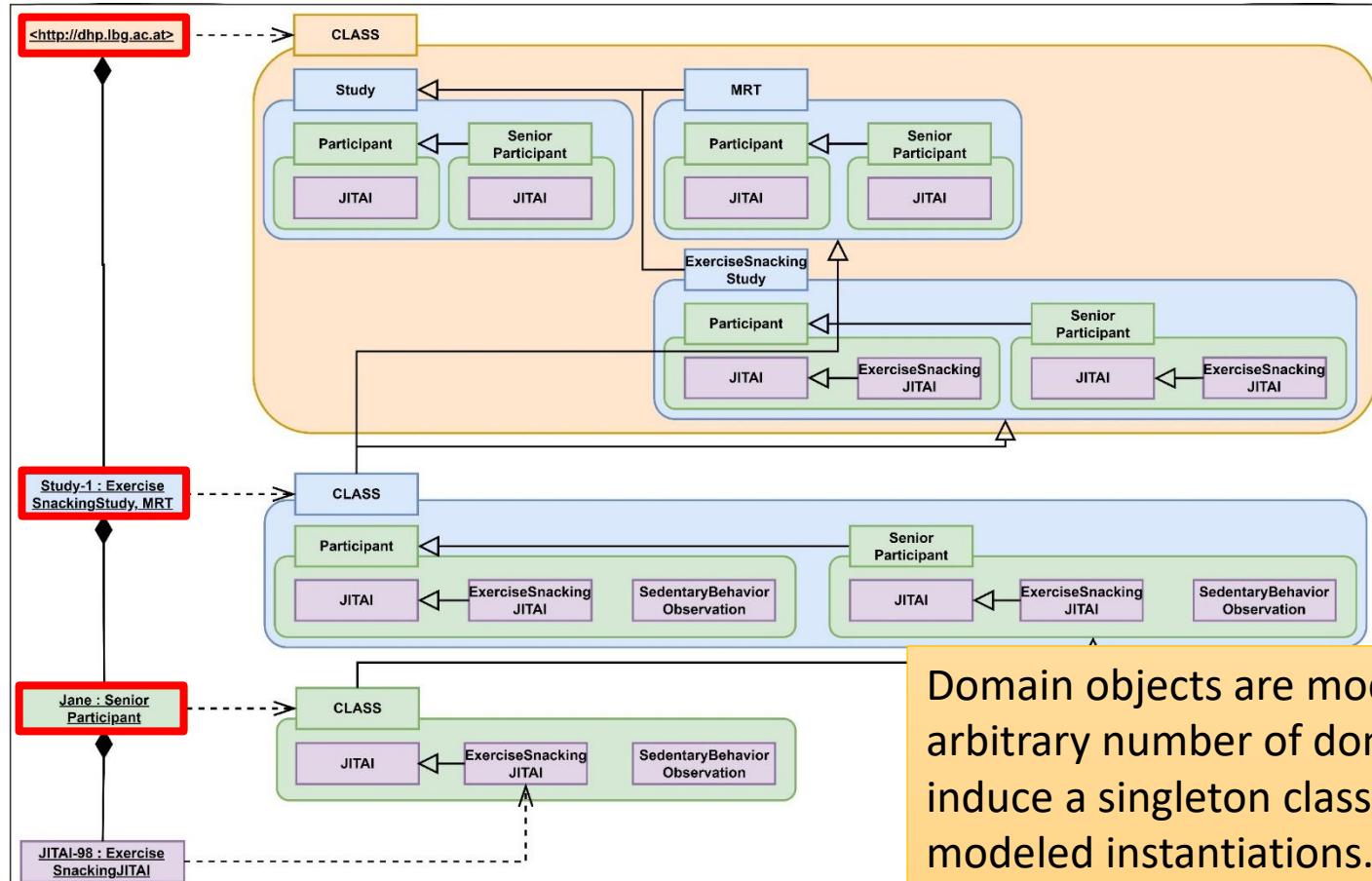
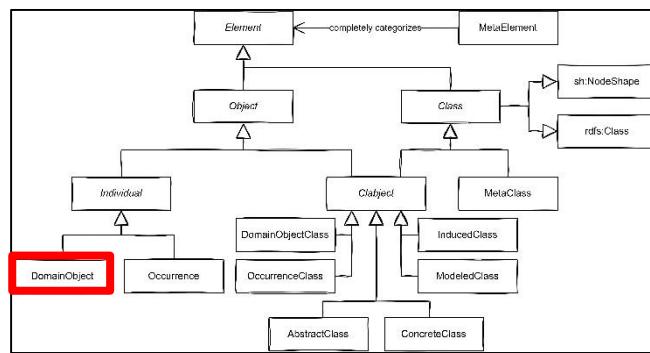




Linguistic Model

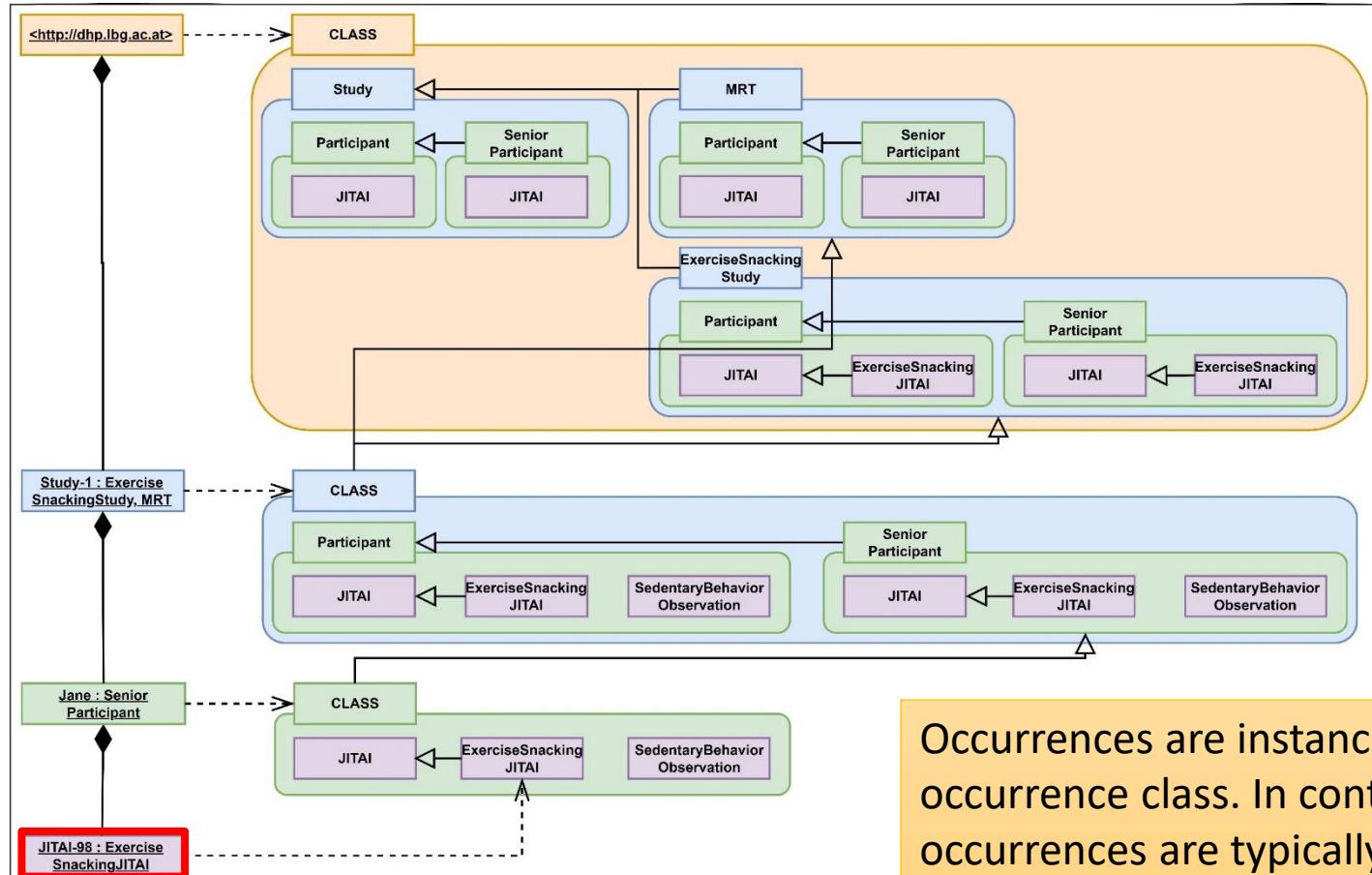
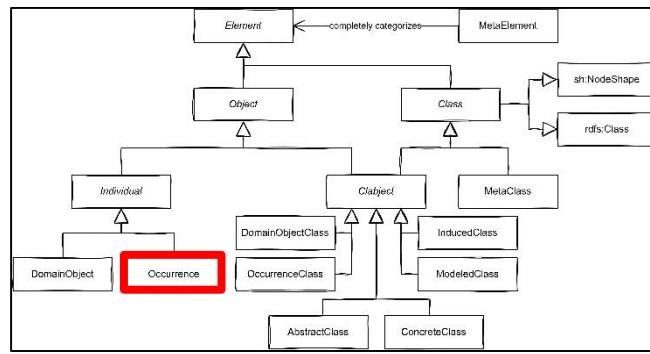


Domain Object



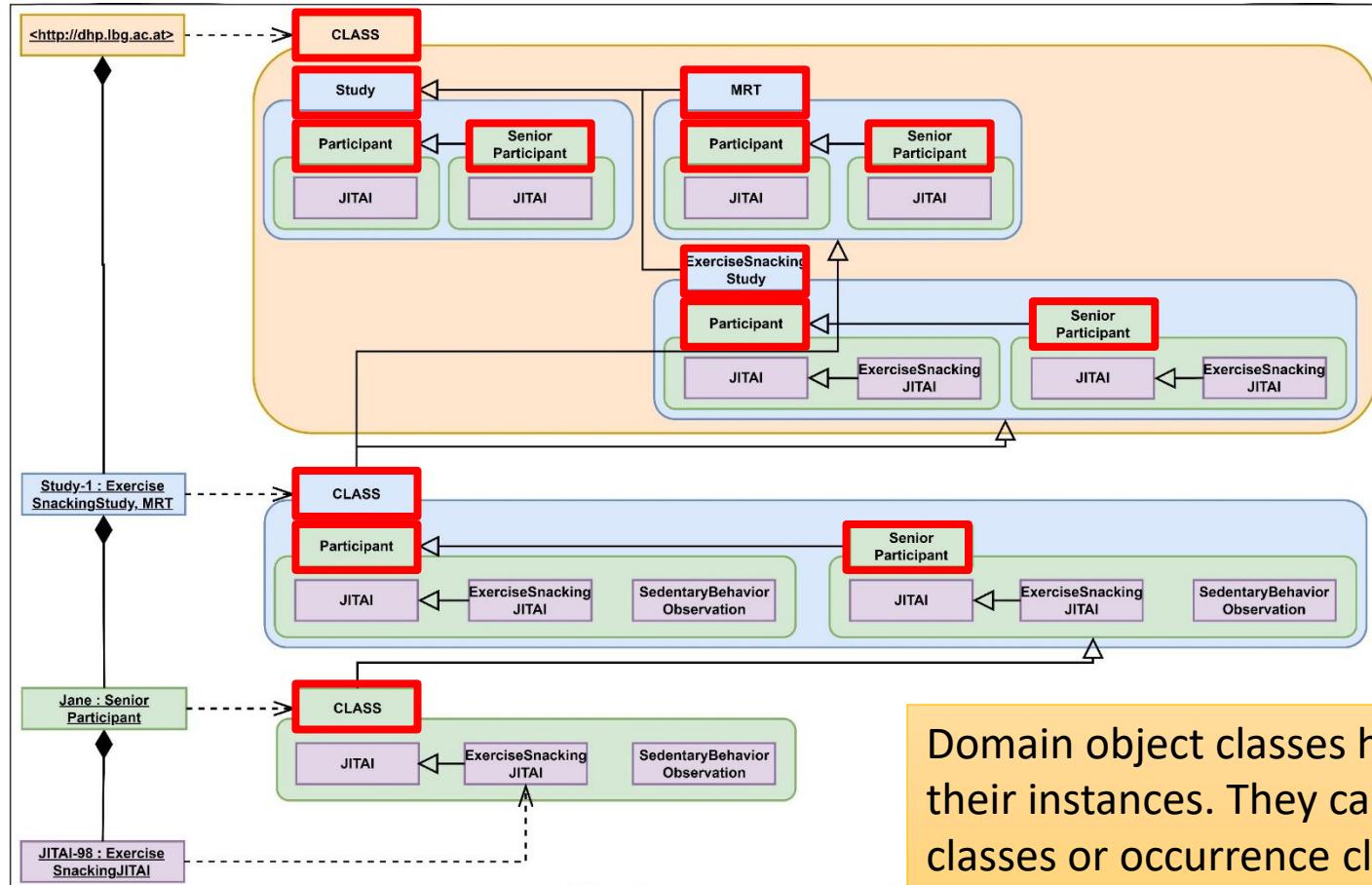
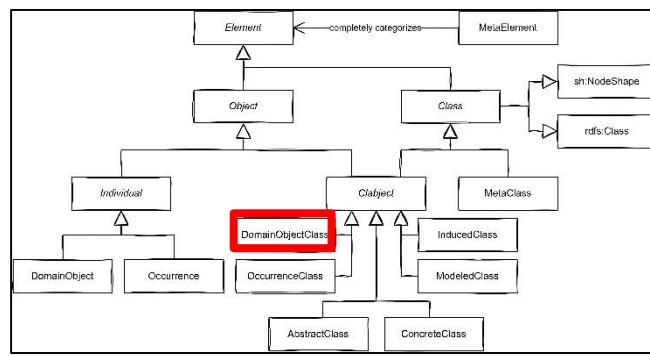
Domain objects are modeled instances of an arbitrary number of domain object classes. They induce a singleton class which specializes the modeled instantiations. The induced singleton class introduces a schema local to the domain object.

Occurrence



Occurrences are instances of exactly one concrete occurrence class. In contrast to domain objects, occurrences are typically created at runtime, thus, introduce no local schema.

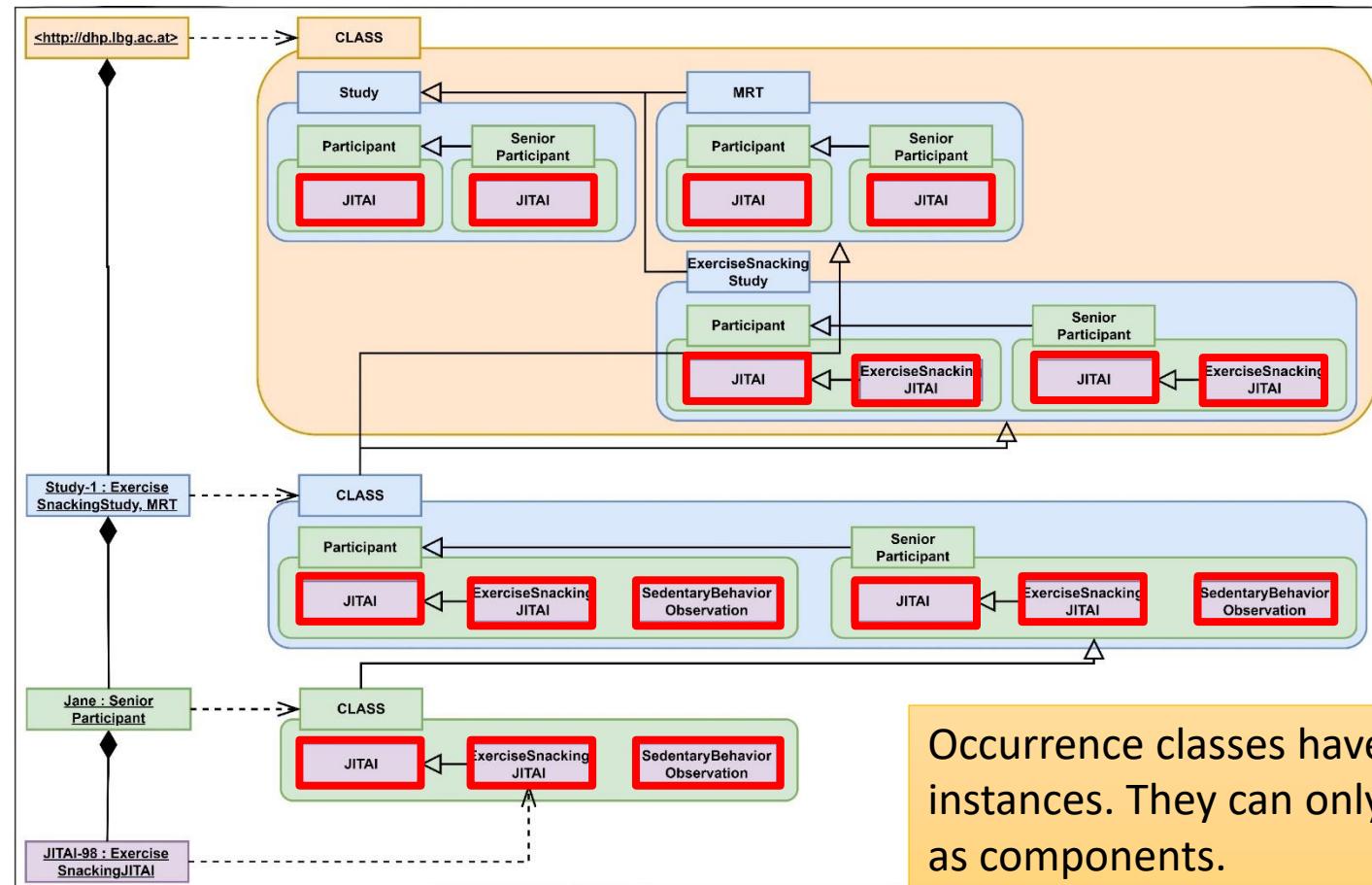
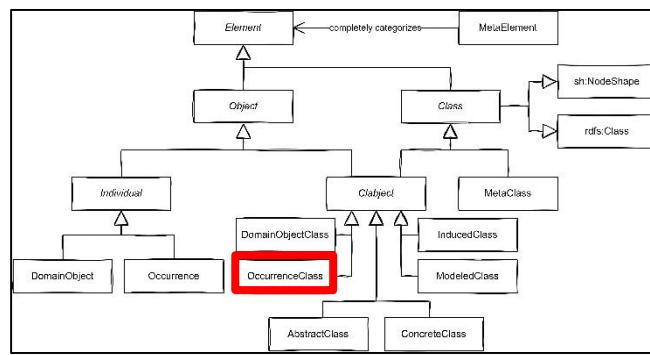
Domain Object Class



Domain object classes have domain objects as their instances. They can have domain object classes or occurrence classes as components.

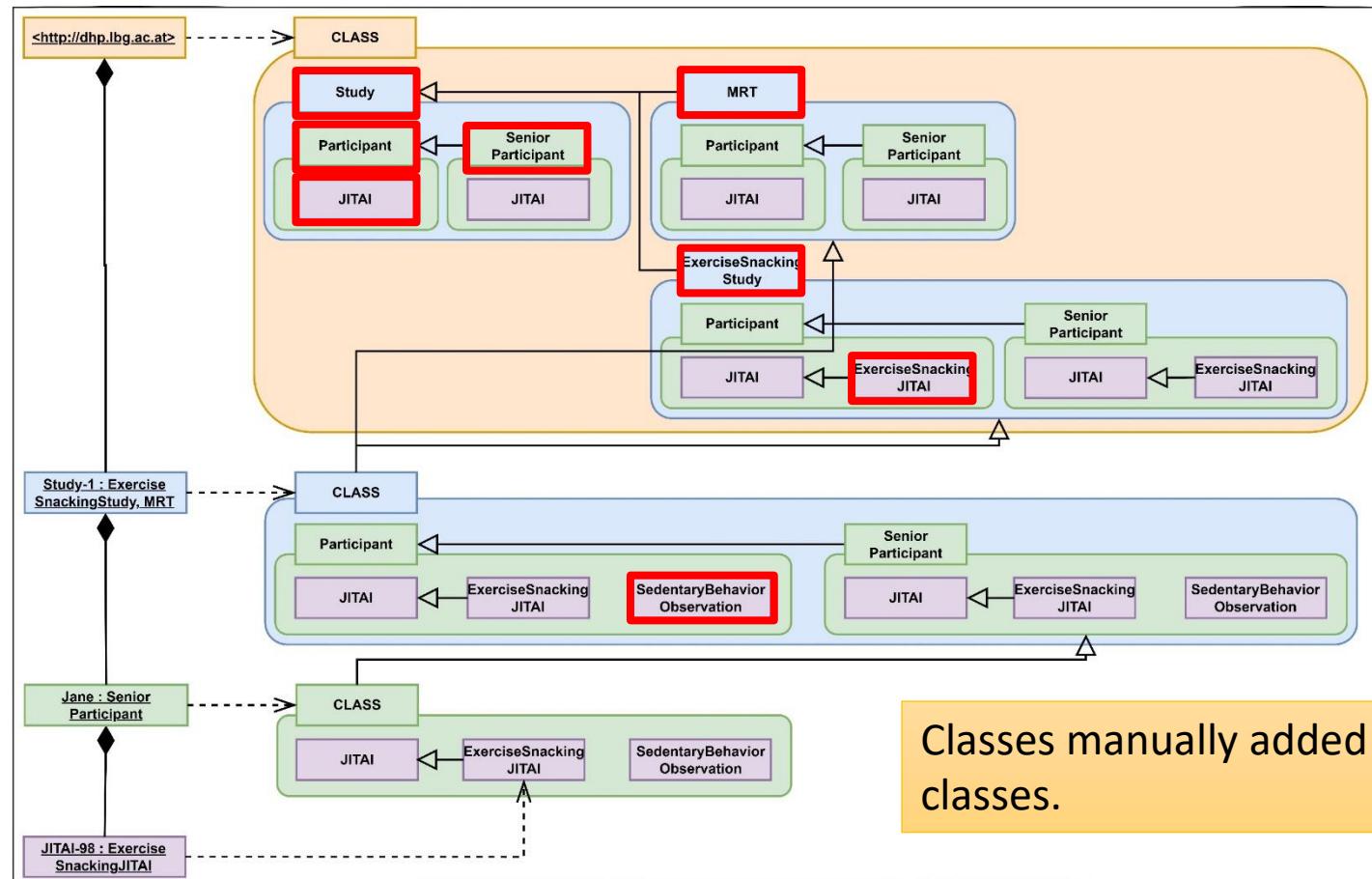
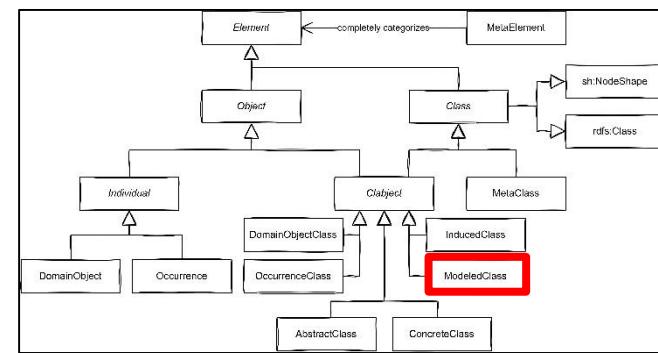
Note: Singleton classes are a combination of domain object classes, concrete classes and induced classes.

Occurrence Class

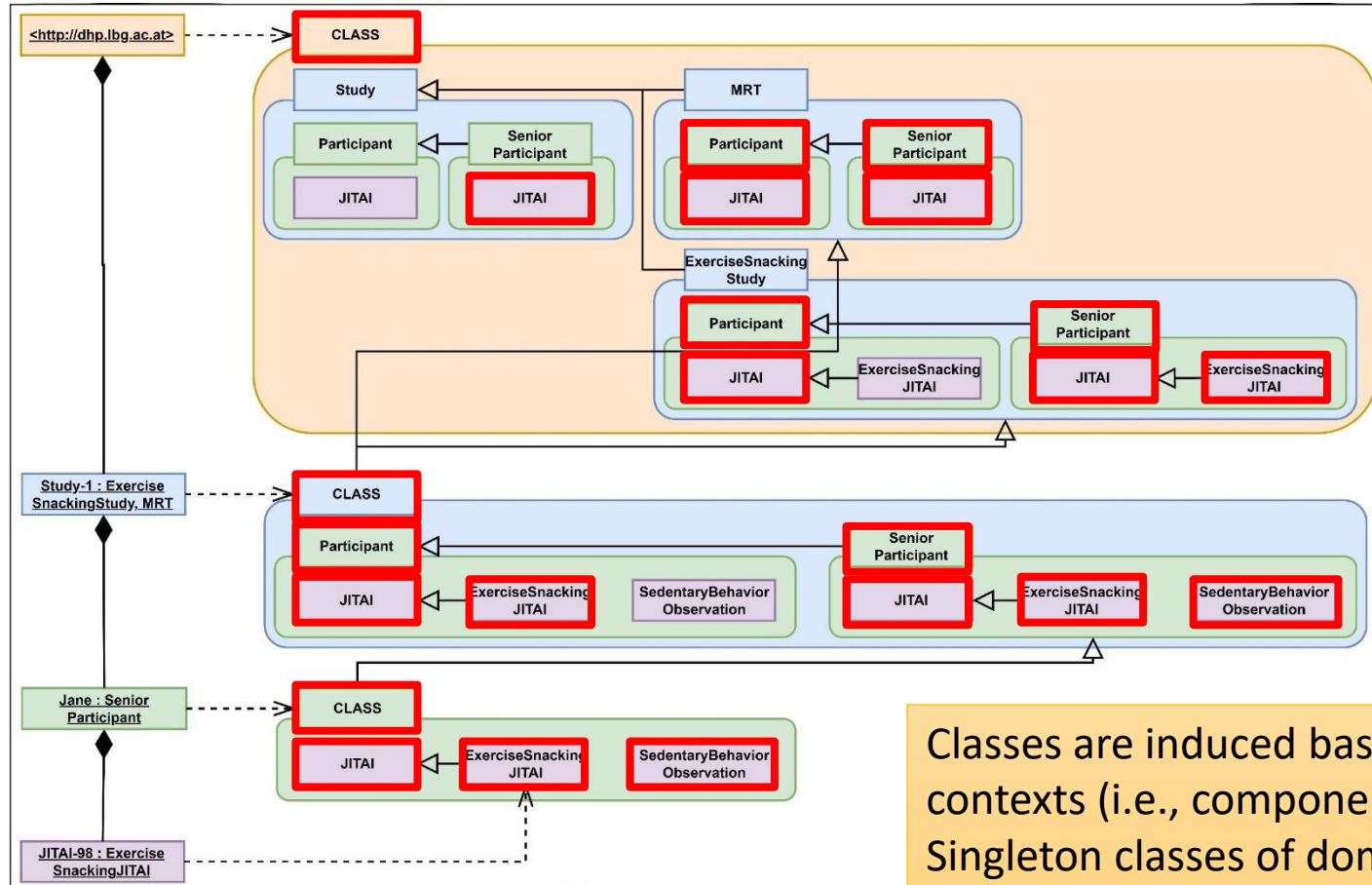
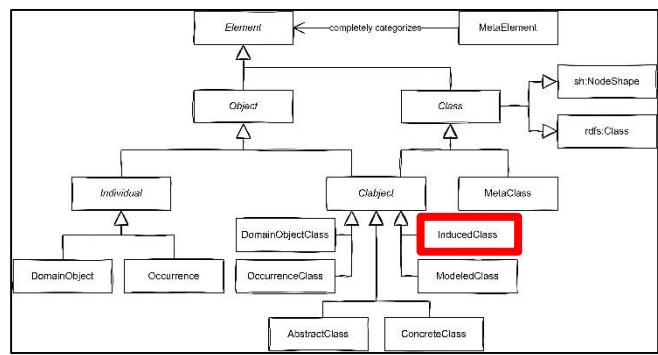


Occurrence classes have occurrences as their instances. They can only have occurrence classes as components.

Modeled Class

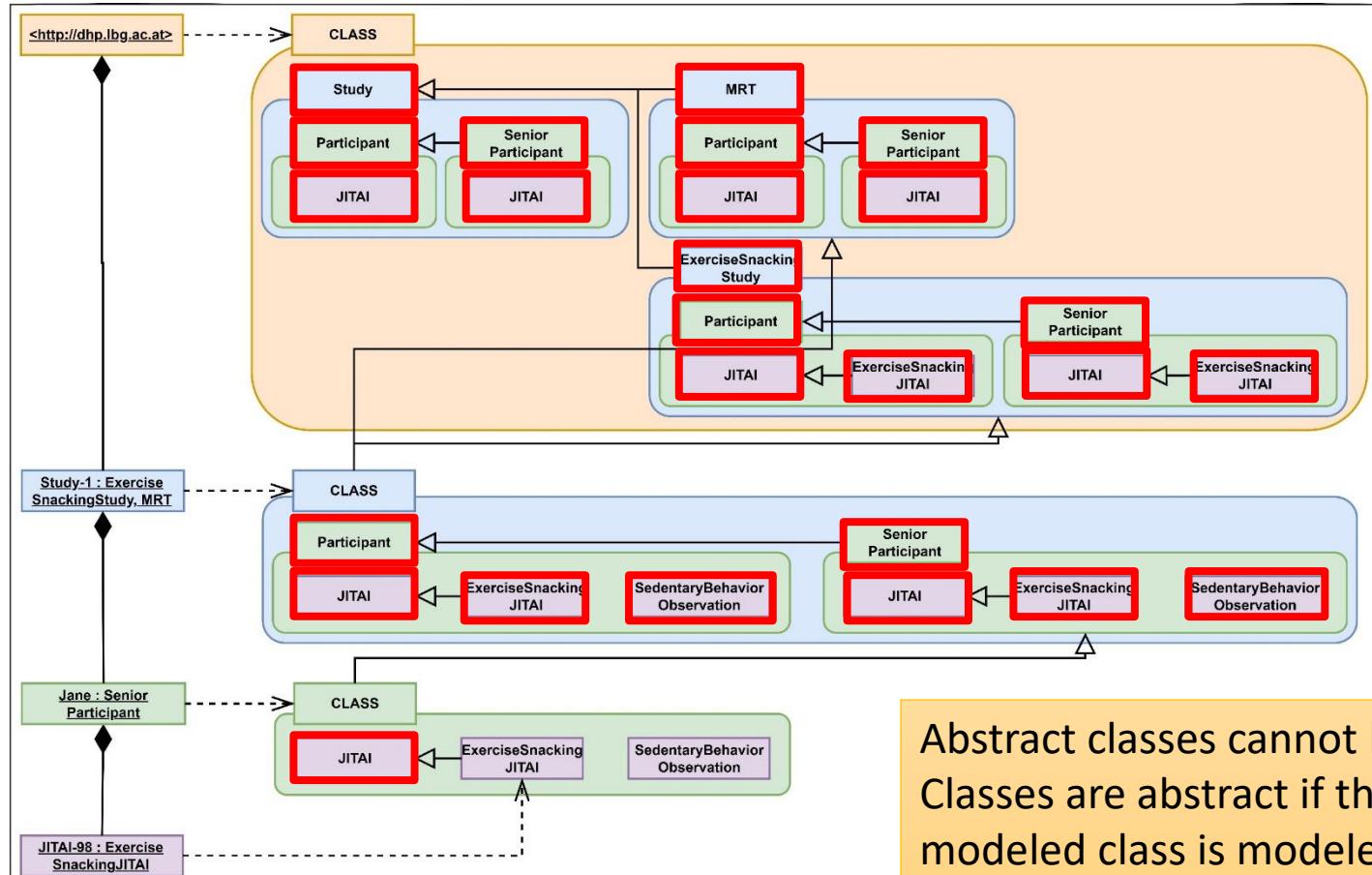
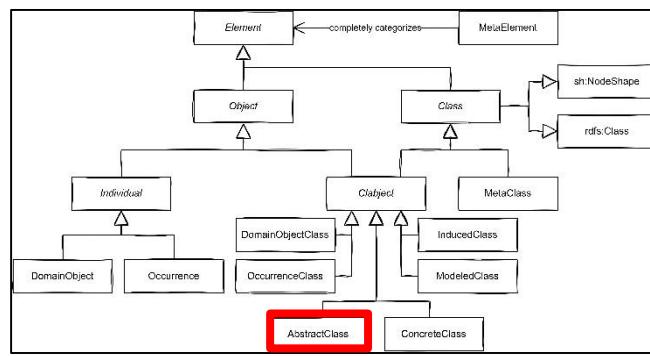


Induced Class

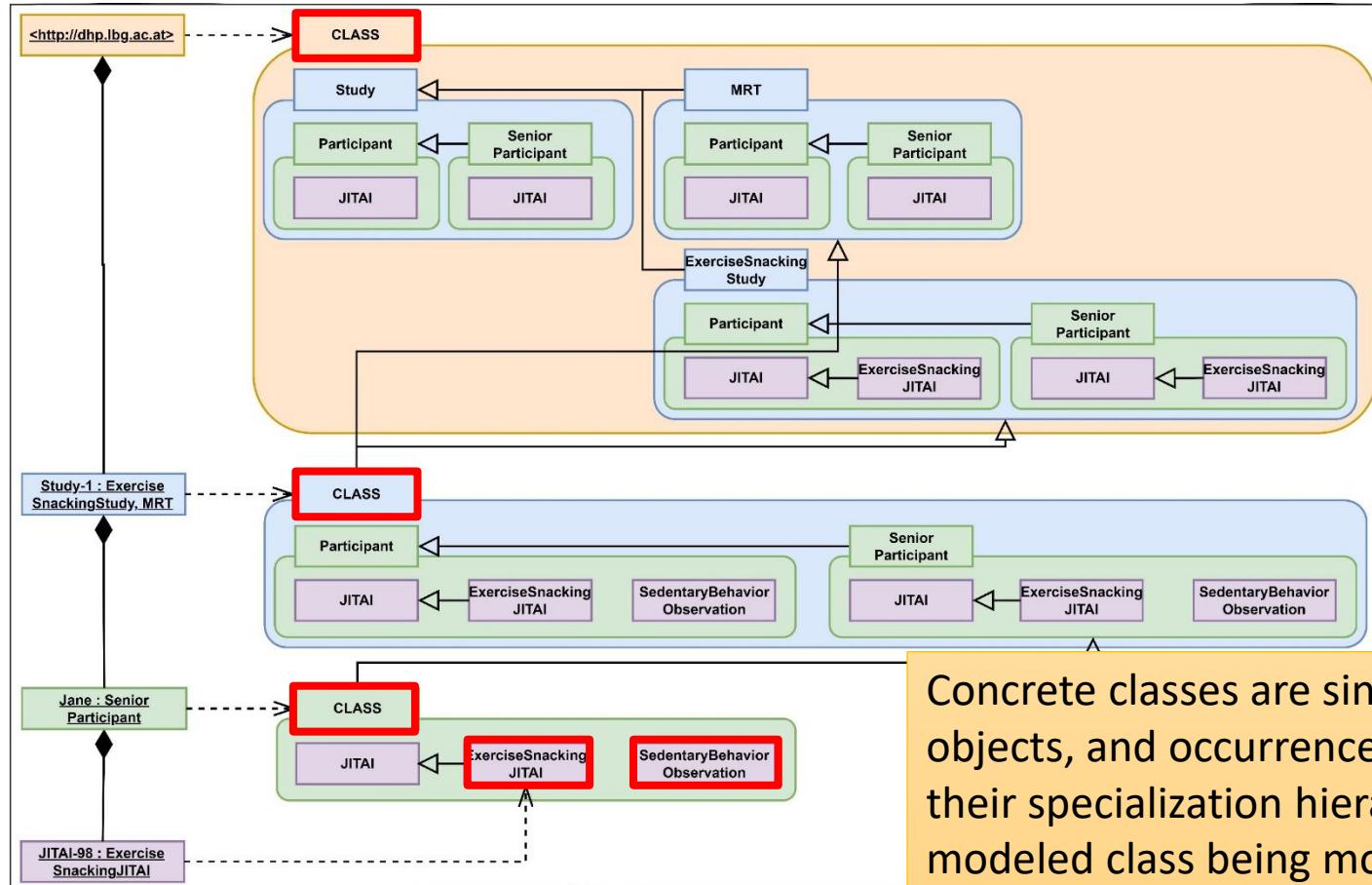
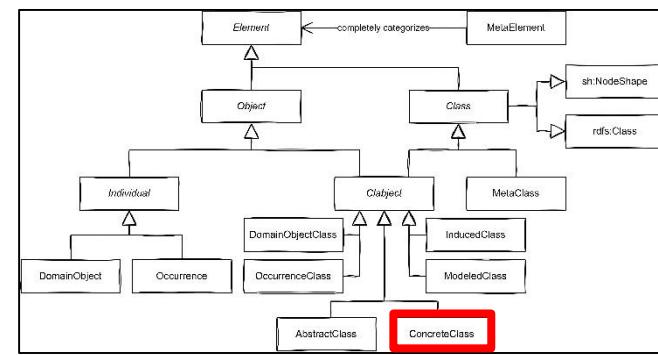


Classes are induced based on classes in super contexts (i.e., components of super classes). Singleton classes of domain objects are induced as well.

Abstract Class

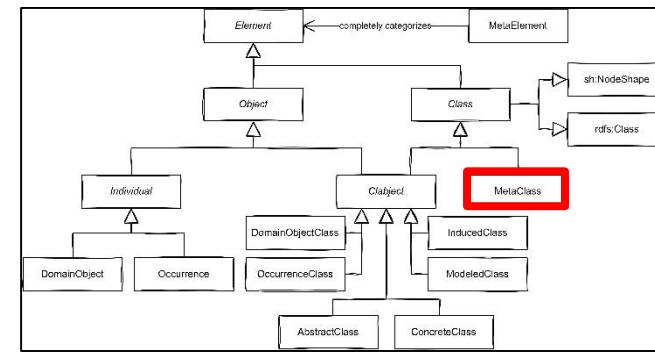
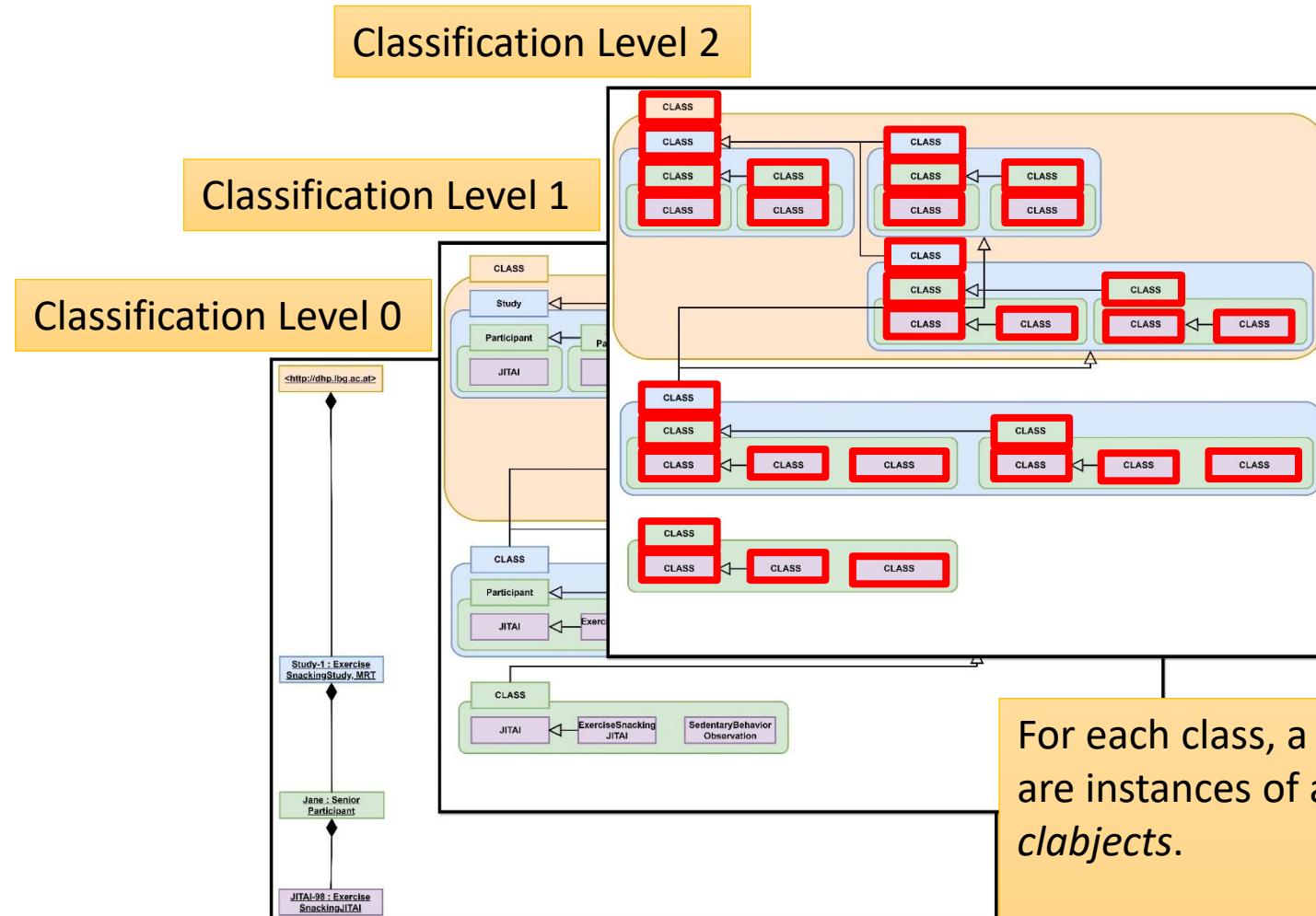


Concrete Class



Concrete classes are singleton classes of domain objects, and occurrence classes that are leafs of their specialization hierarchy and with their modeled class being modeled as concrete class. Concrete classes cannot be specialized. Individuals directly instantiate concrete classes.

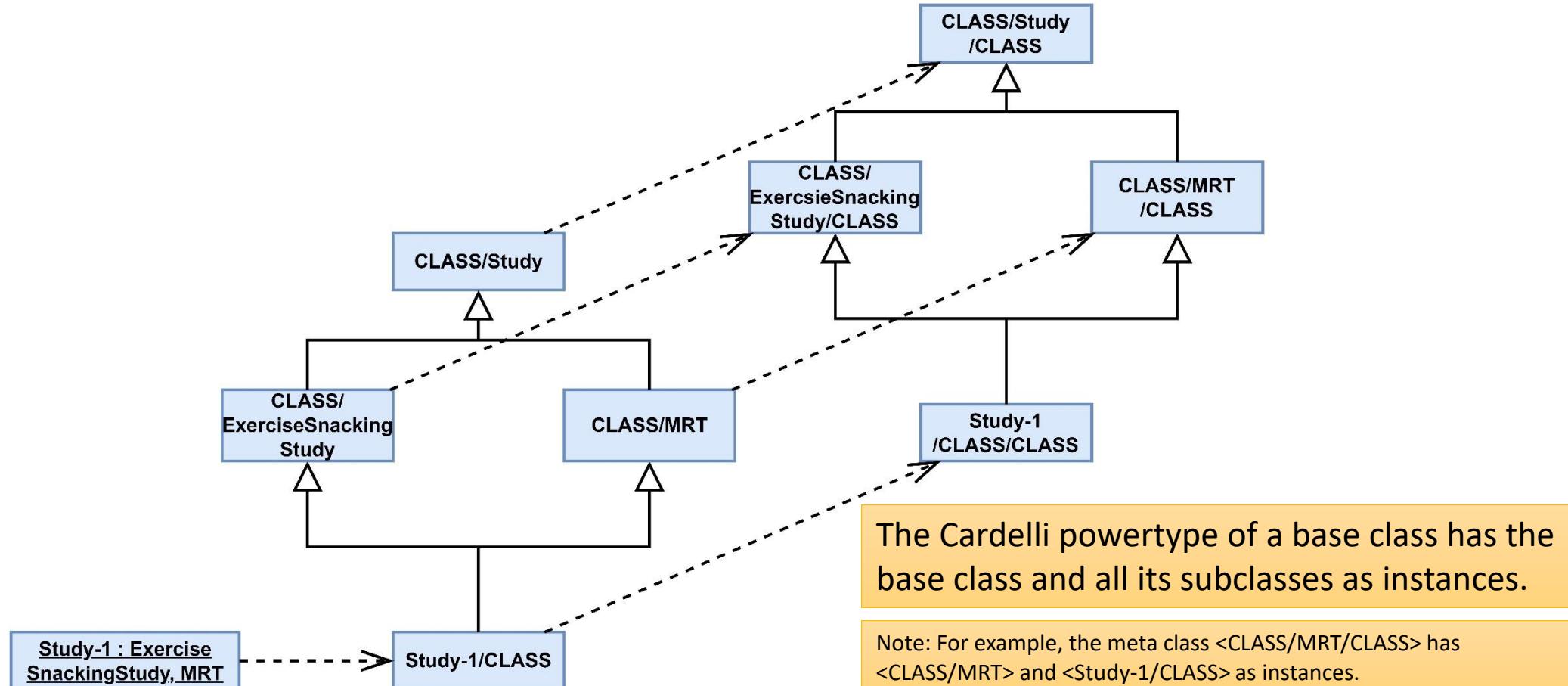
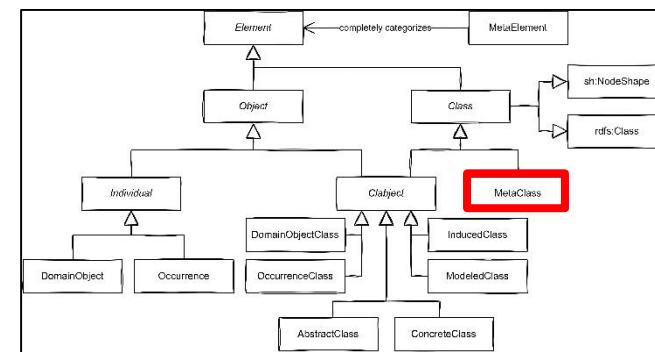
Meta Class



For each class, a meta class is derived. Classes that are instances of a meta class are referred to as *clobjects*.

The meta class corresponds to the Cardelli powertype.

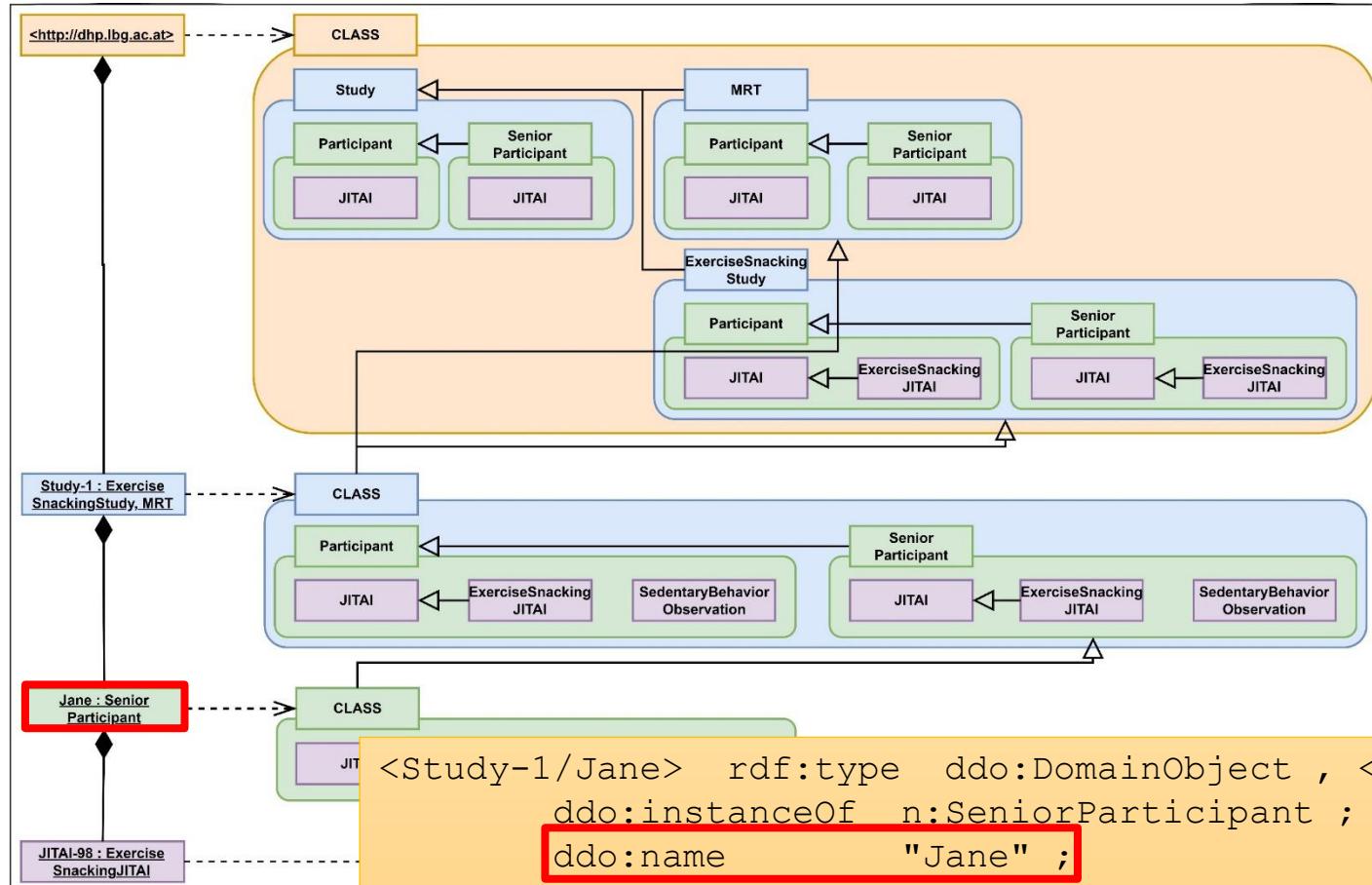
Cardelli Powertype



Properties

ddo:name

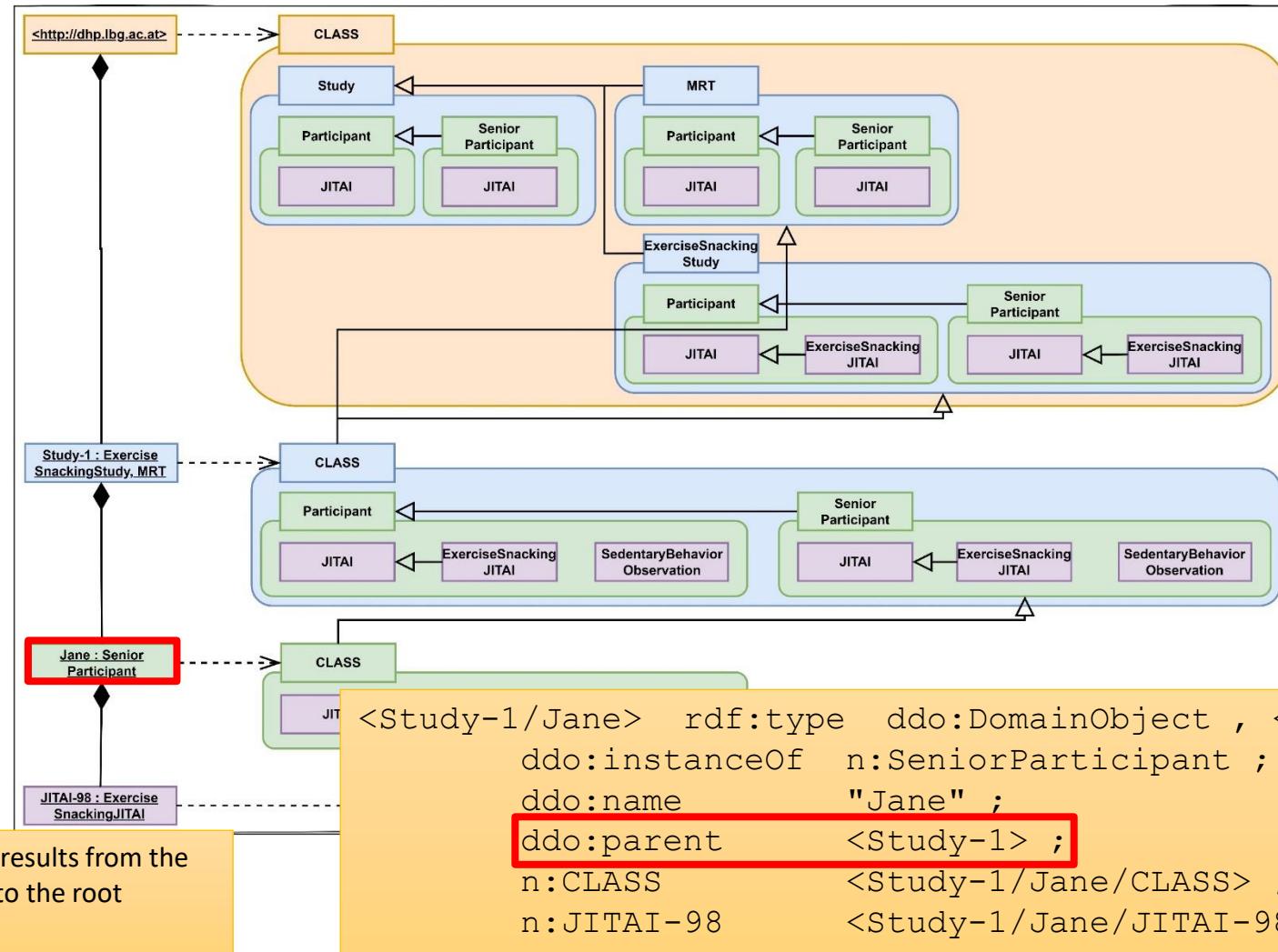
The name of a DDO element is visualized as text in the rectangles. It is unique under a composite element. Modeled classes introduce a name that is propagated to its induced classes. Singleton classes and meta classes are named "CLASS".



```
<Study-1/Jane> rdf:type ddo:DomainObject , <Study-1/Jane/CLASS> ;  
    ddo:instanceOf n:SeniorParticipant ;  
    ddo:name "Jane" ;  
    ddo:parent <Study-1> ;  
    n:CLASS <Study-1/Jane/CLASS> ;  
    n:JITAI-98 <Study-1/Jane/JITAI-98> .
```

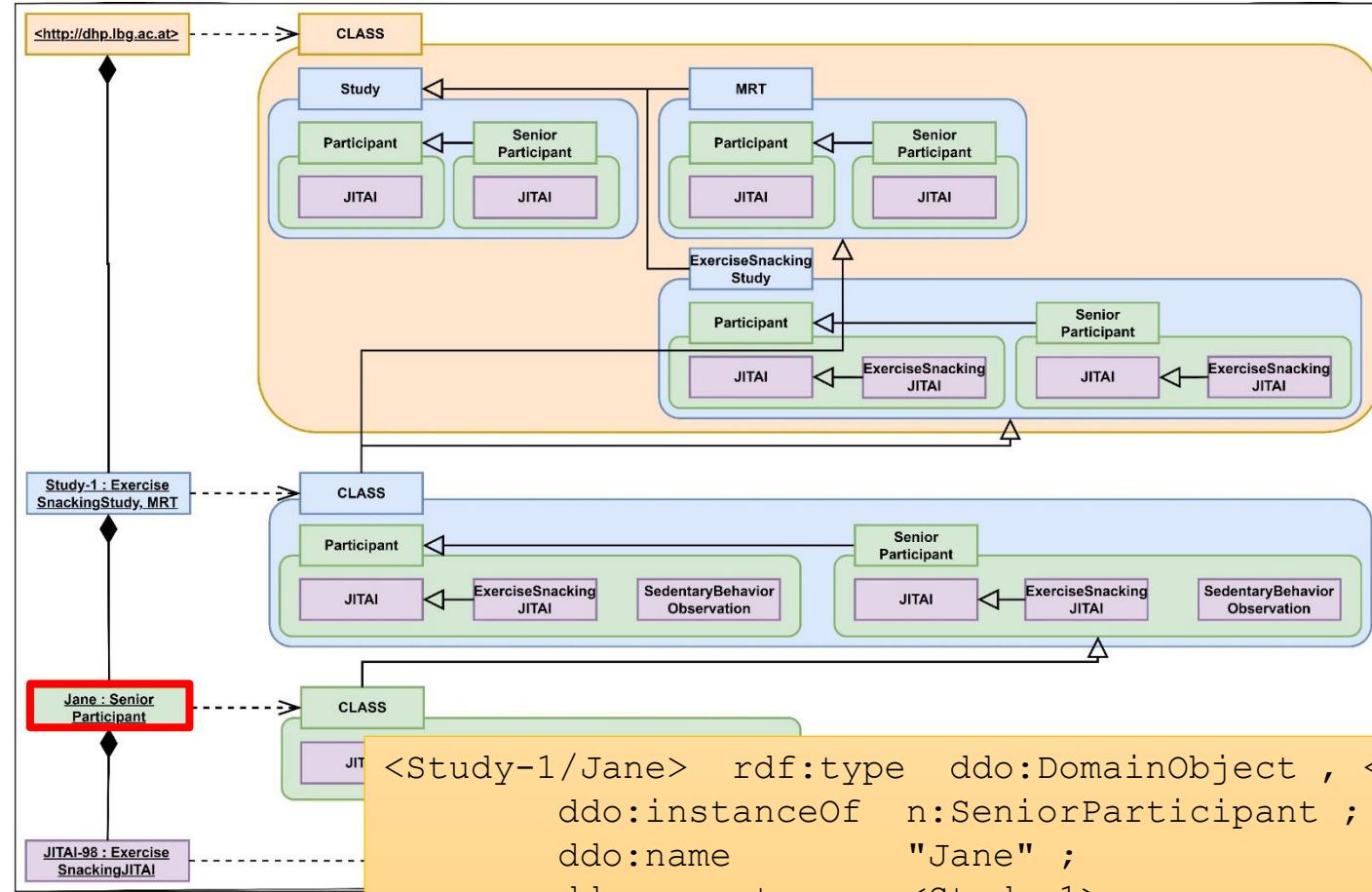
ddo:parent

Objects and classes have their composite as parent. Induced singleton classes and meta classes have their object as parent.



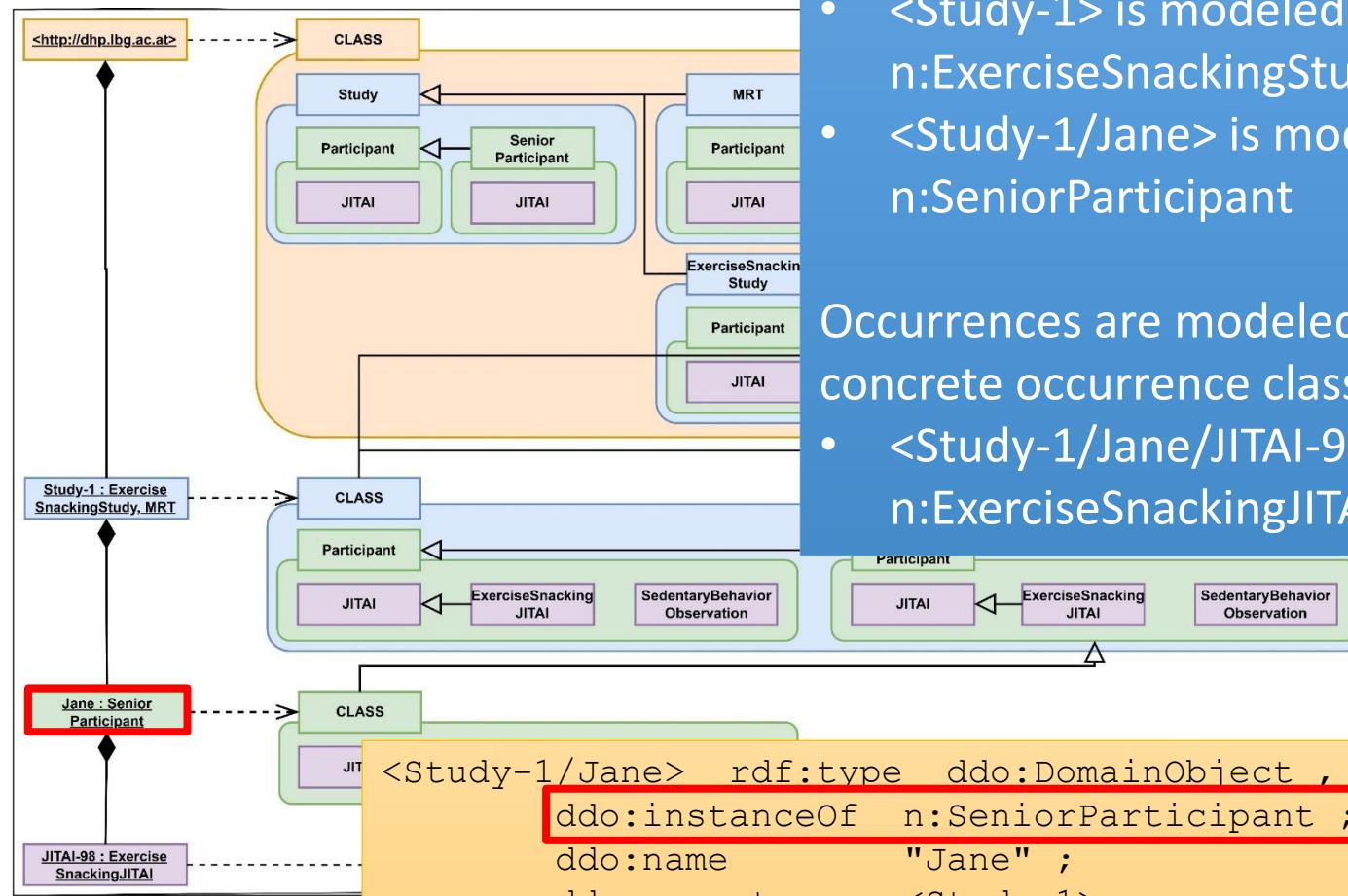
Named Childs

Named child properties are derived as a reverse property of ddo:parent and combined with ddo:name of the element.



```
<Study-1/Jane> rdf:type ddo:DomainObject , <Study-1/Jane/CLASS> ;  
    ddo:instanceOf n:SeniorParticipant ;  
    ddo:name "Jane" ;  
    ddo:parent <Study-1> ;  
    n:CLASS <Study-1/Jane/CLASS> ;  
    n:JITAI-98 <Study-1/Jane/JITAI-98> .
```

ddo:instanceOf



Individuals are modeled instances of classes.
Domain objects can be modeled as instance of arbitrary classes:

- <<http://dhp.lbg.ac.at>> has no modeled instantiation
- <Study-1> is modeled instance of n:ExerciseSnackingStudy and n:MRT
- <Study-1/Jane> is modeled instance of n:SeniorParticipant

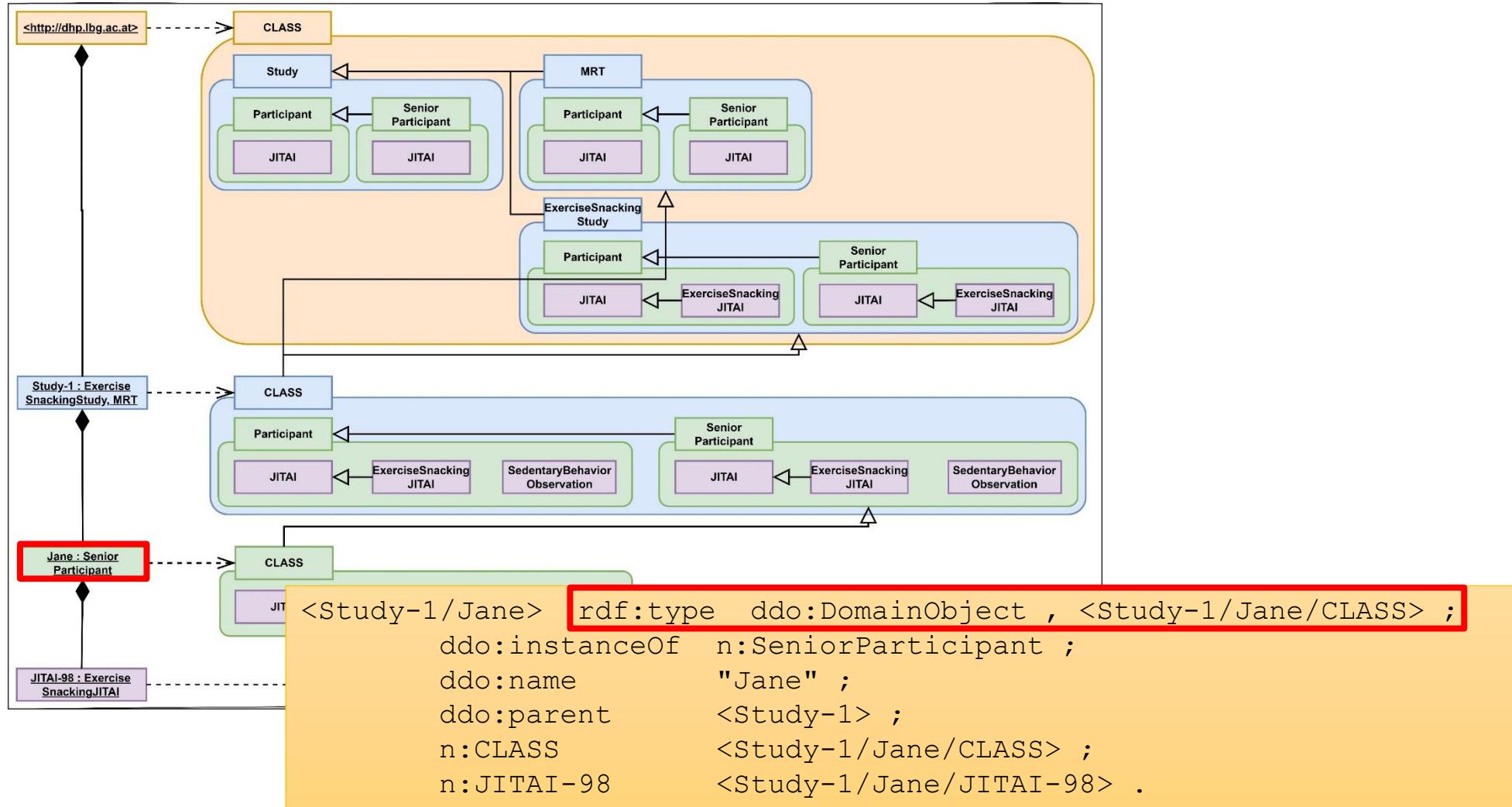
Occurrences are modeled instance of exactly one concrete occurrence class:

- <Study-1/Jane/JITA1-98> is modeled instance of n:ExerciseSnackingJITAI

```
<Study-1/Jane> rdf:type ddo:DomainObject , <Study-1/Jane/CLASS> ;  
      ddo:instanceOf n:SeniorParticipant ;  
      ddo:name "Jane" ;  
      ddo:parent <Study-1> ;  
      n:CLASS <Study-1/Jane/CLASS> ;  
      n:JITA1-98 <Study-1/Jane/JITA1-98> .
```

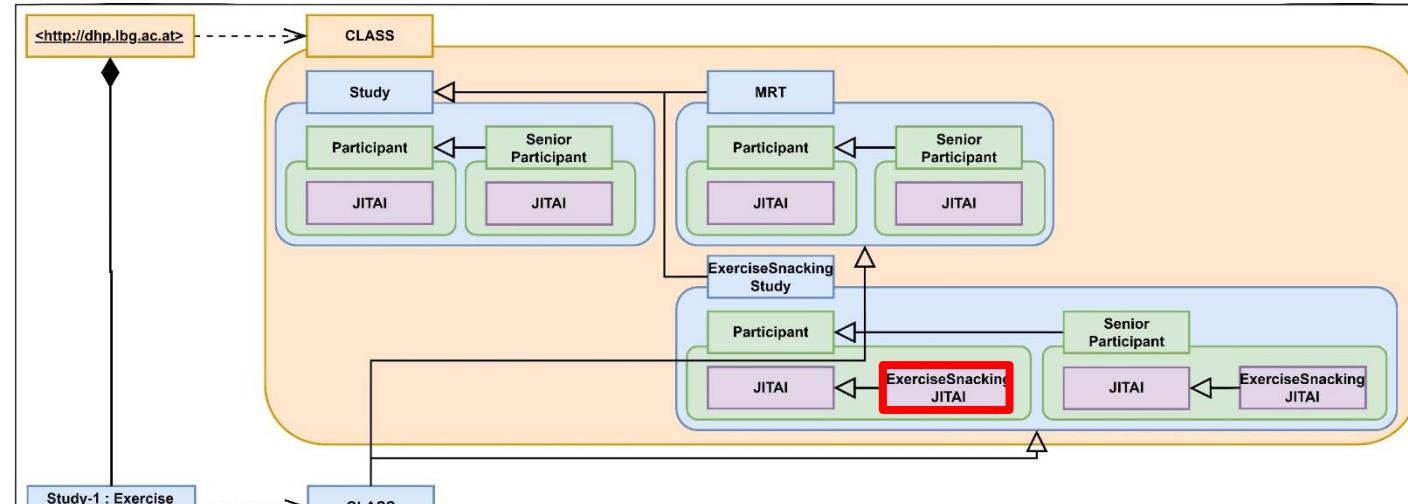
`rdf:type`

Objects are instances of classes. The linguistic model is instantiated orthogonally. The materialized `rdf:type` is used as a direct instance of relationship for both kinds of instantiation.



Modeled sub classes are specializations of at least one class under the same composition level.

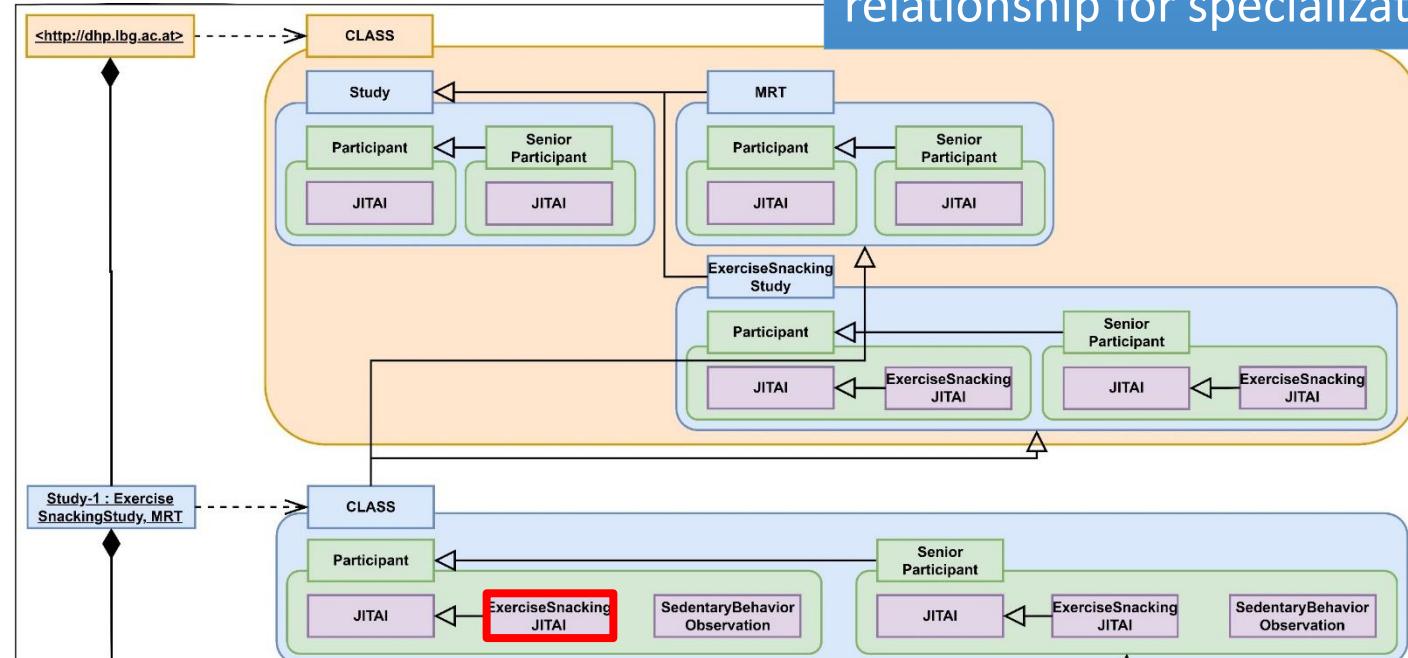
ddo:specializationOf



```
<CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI>
    rdf:type ddo:ModeledClass , ddo:AbstractClass , ddo:OccurrenceClass ,
    <CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI/CLASS> ;
    <CLASS/ExerciseSnackingStudy/Participant/JITAI> ;
    ddo:ConcreteClass ;
    <CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI> ;
    "ExerciseSnackingJITAI" ;
    <CLASS/ExerciseSnackingStudy/Participant> ;
    ddo:specializationOf n:JITAI ;
    n:CLASS <CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI/CLASS> .
```

rdfs:subClassOf

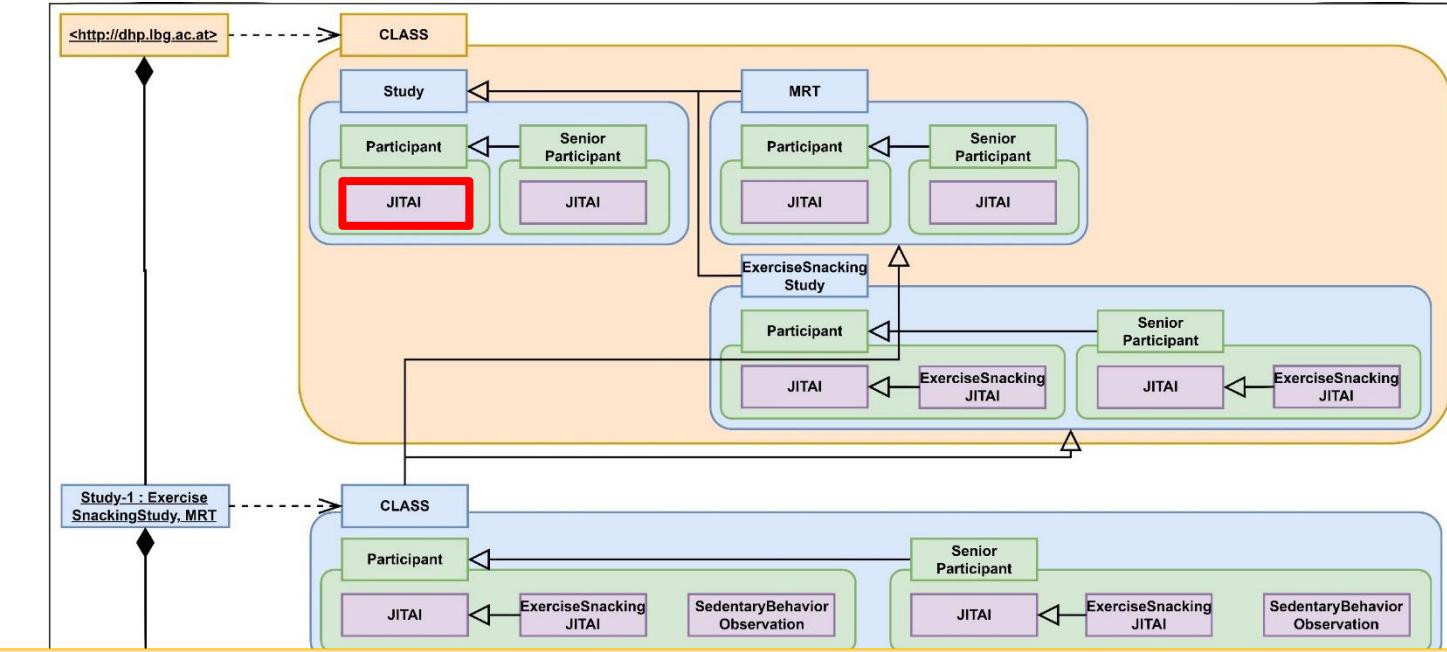
Classes (i.e., clabjects and meta classes) can be sub classes of other classes under the same composition level or under the composition level of the composite's super classes. The materialized rdfs:subClassOf is used as a direct sub class of relationship for specializations.



```
<Study-1/CLASS/Participant/ExerciseSnackingJITAI>
    rdf:type          ddo:AbstractClass , ddo:InducedClass , ddo:OccurrenceClass ,
                  <Study-1/CLASS/Participant/ExerciseSnackingJITAI/CLASS> ;
    rdfs:subClassOf   <CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI> ,
                      <Study-1/CLASS/Participant/JITAI> ;
    ddo:modeledClass <CLASS/ExerciseSnackingStudy/Participant/ExerciseSnackingJITAI> ;
    ddo:name         "ExerciseSnackingJITAI" ;
    ddo:parent        <Study-1/CLASS/Participant> ;
    n:CLASS          <Study-1/CLASS/Participant/ExerciseSnackingJITAI/CLASS> .
```

ddo:leafs

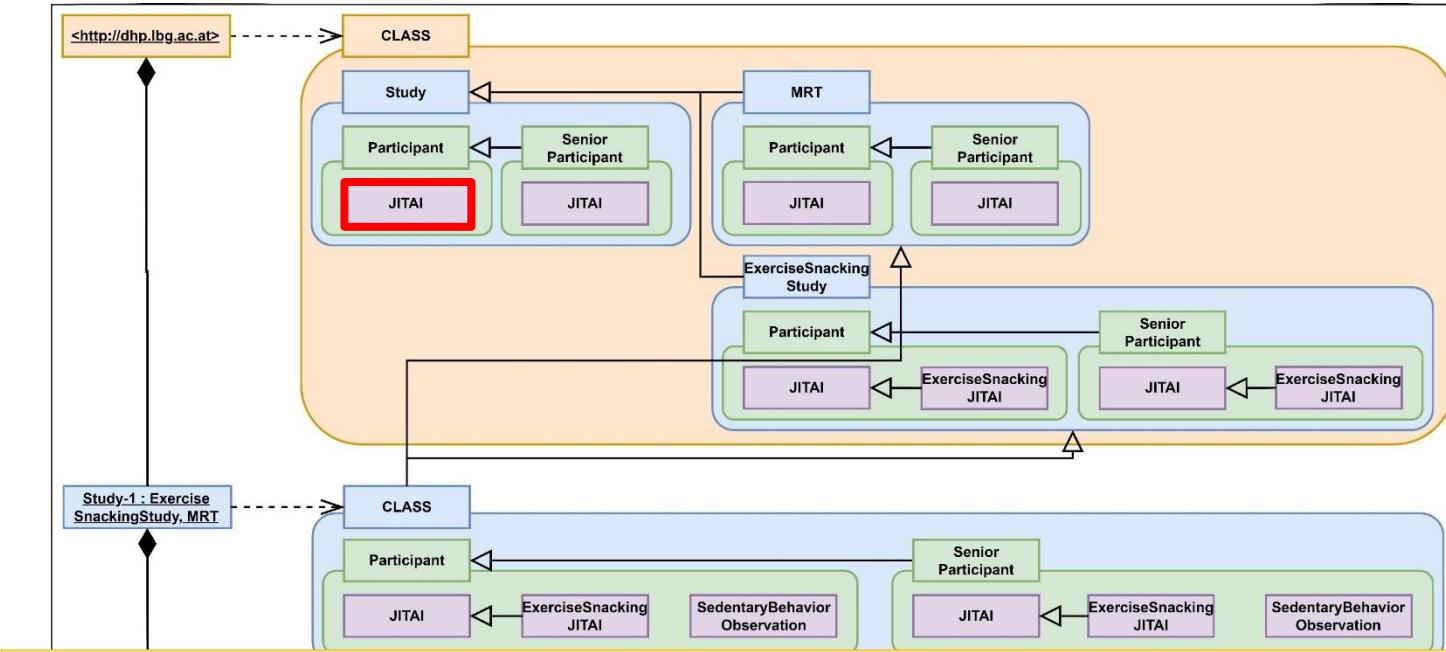
When modeling occurrence classes, it is already determined whether leafs of that class are abstract or concrete classes. Classes modeled with leafs as concrete classes cannot be further specialized.



```
<CLASS/Study/Participant/JITAI>
    rdf:type          ddo:OccurrenceClass , ddo:ModeledClass ,
                    <CLASS/Study/Participant/JITAI/CLASS> , ddo:AbstractClass ;
    ddo:leafs        ddo:AbstractClass ;
    ddo:modeledClass <CLASS/Study/Participant/JITAI> ;
    ddo:name         "JITAI" ;
    ddo:parent       <CLASS/Study/Participant> ;
    n:CLASS          <CLASS/Study/Participant/JITAI/CLASS> .
```

ddo:modeledClass (1)

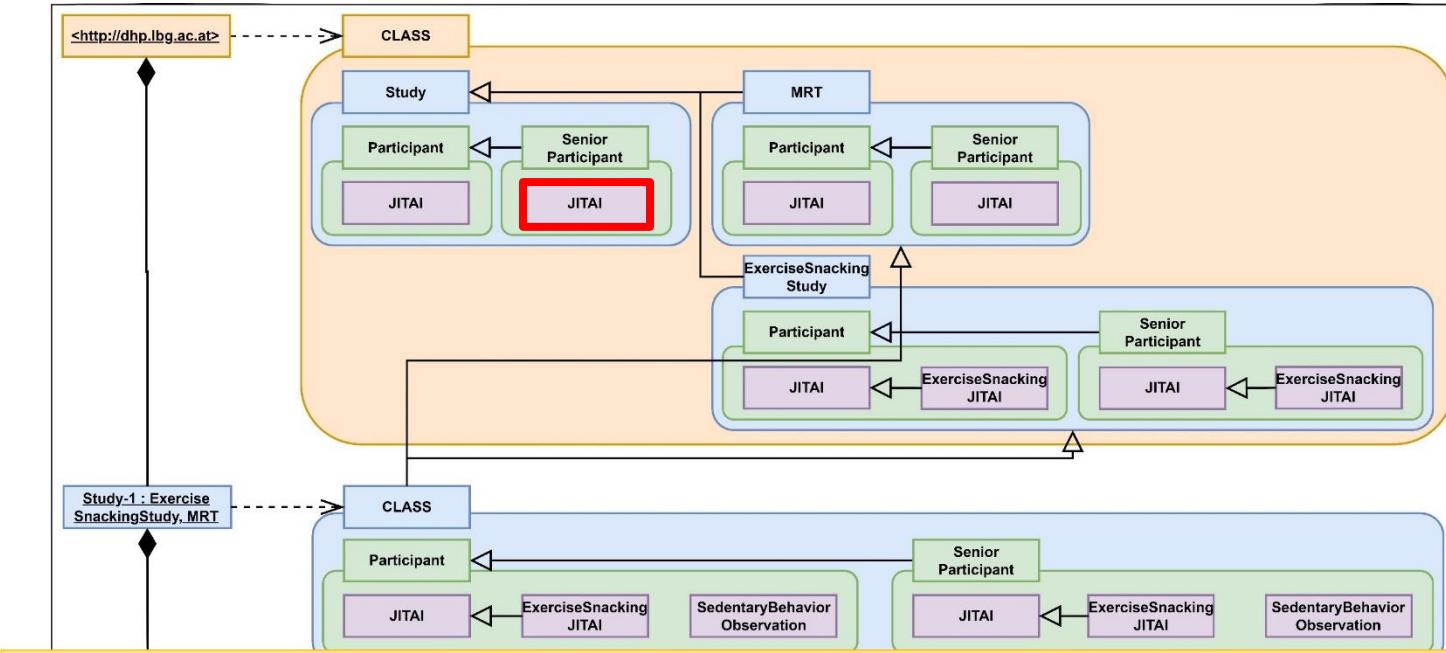
Clabjects refer to their modeled class. Modeled classes refer to themselves, while singleton classes do not refer to modeled classes (they are not induced because of a modeled class).



```
<CLASS/Study/Participant/JITAI>
    rdf:type          ddo:OccurrenceClass , ddo:ModeledClass ,
                    <CLASS/Study/Participant/JITAI/CLASS> , ddo:AbstractClass ;
    ddo:leafs         ddo:AbstractClass ;
    ddo:modeledClass <CLASS/Study/Participant/JITAI> ;
    ddo:name          "JITAI" ;
    ddo:parent        <CLASS/Study/Participant> ;
    n:CLASS           <CLASS/Study/Participant/JITAI/CLASS> .
```

ddo:modeledClass (2)

Clabjects refer to their modeled class. Modeled classes refer to themselves, while singleton classes do not refer to modeled classes (they are not induced because of a modeled class).



```
<CLASS/Study/SeniorParticipant/JITAI>
    rdf:type          ddo:OccurrenceClass , ddo:InducedClass , ddo:AbstractClass ,
                  <CLASS/Study/SeniorParticipant/JITAI/CLASS> ;
    rdfs:subClassOf   <CLASS/Study/Participant/JITAI> ;
    ddo:modeledClass  <CLASS/Study/Participant/JITAI> ;
    ddo:name          "JITAI" ;
    ddo:parent         <CLASS/Study/SeniorParticipant> ;
    n:CLASS           <CLASS/Study/SeniorParticipant/JITAI/CLASS> .
```

Live Demo of DDO-Playground

DDO Playground

- You can download the DDO-Playground here
 - <https://resources.lbi-dhp.at/mlm4jtais/ddo-playground-v1.jar>
- You may have to install Oracle JDK 18.
- You can run the playground from command line with the following command: `java –jar ddo-playground-v1.jar`

Task

For Task (1) – (6) please save and submit your final domain model.

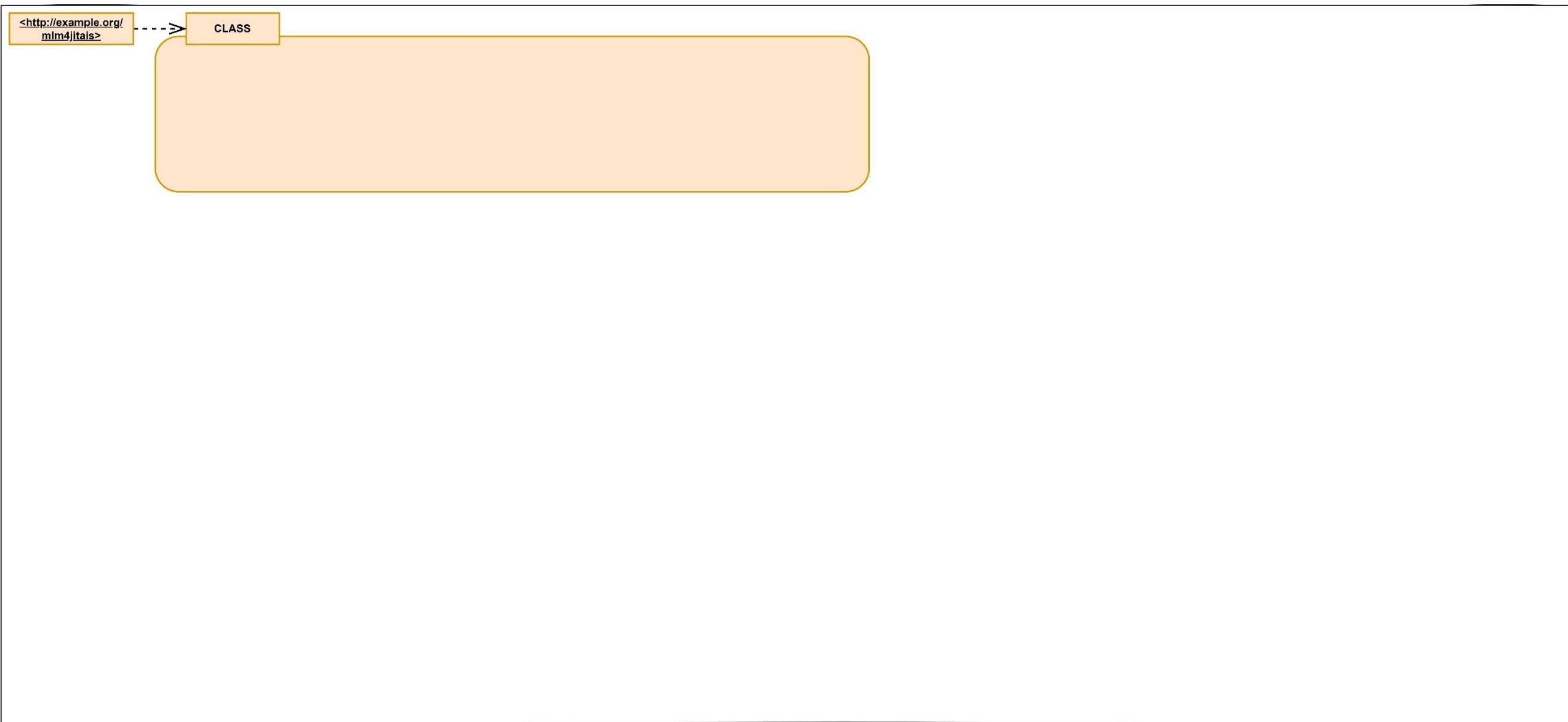
For Task (7) – (10) please submit your SPARQL Queries.

Task (1)

In this task, you create a domain model for developing and evaluating activity and dietary JITAls in multiple studies.

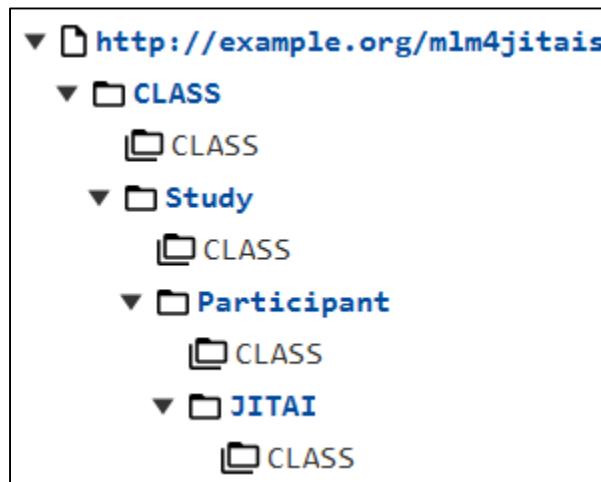
- Create a new domain model with the URI
<http://example.org/mlm4jita1s>
- Expected outcome

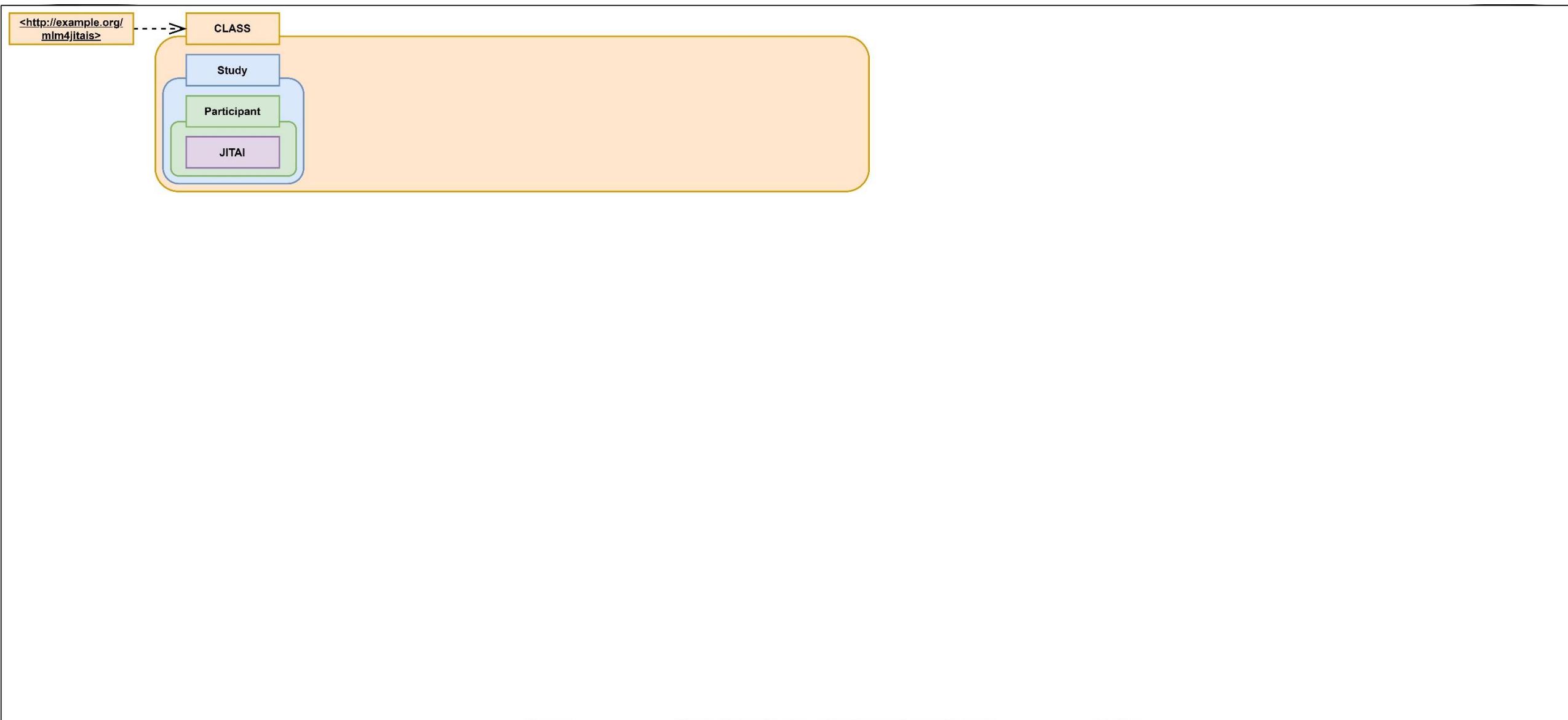




Task (2)

- Add local to that domain model a composition hierarchy with class *JITAI* as component of class *Participant* as component of class *Study*
 - Note: Individuals of classes *Study* and *Participant* are domain objects
 - Note: Individuals of class *JITAI* are occurrences
 - Note: Leafs of class *JITAI* are abstract classes
- Expected outcome

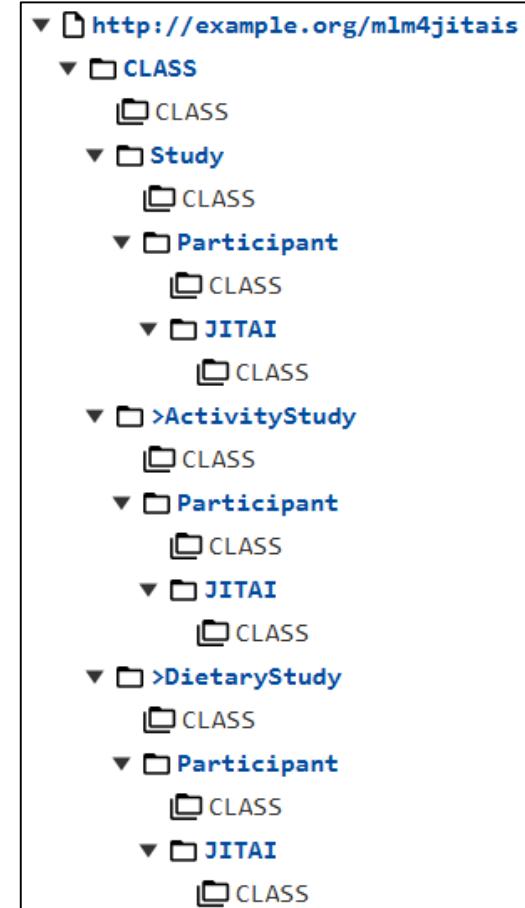


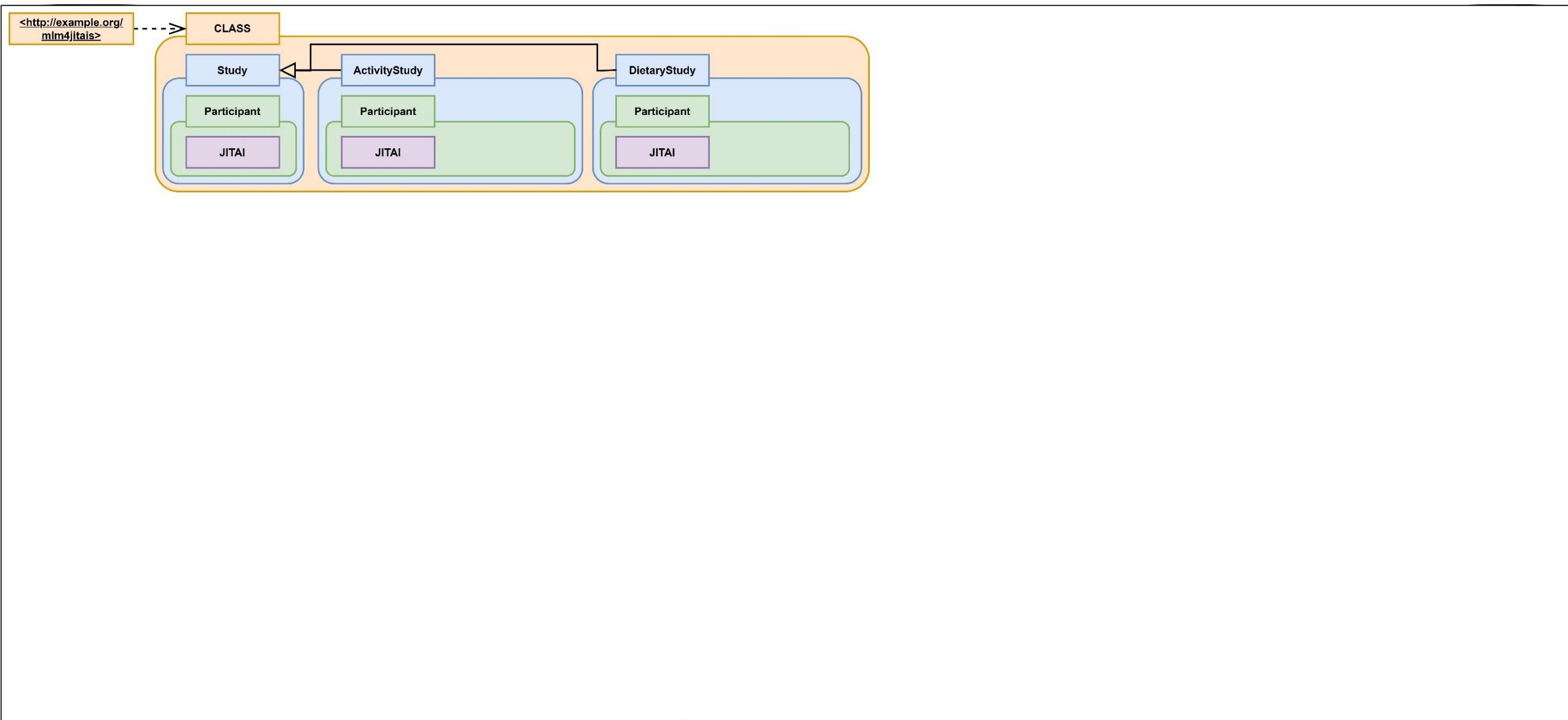


Task (3)

- Add classes *ActivityStudy* and *DietaryStudy* as specializations of class *Study*

Expected outcome

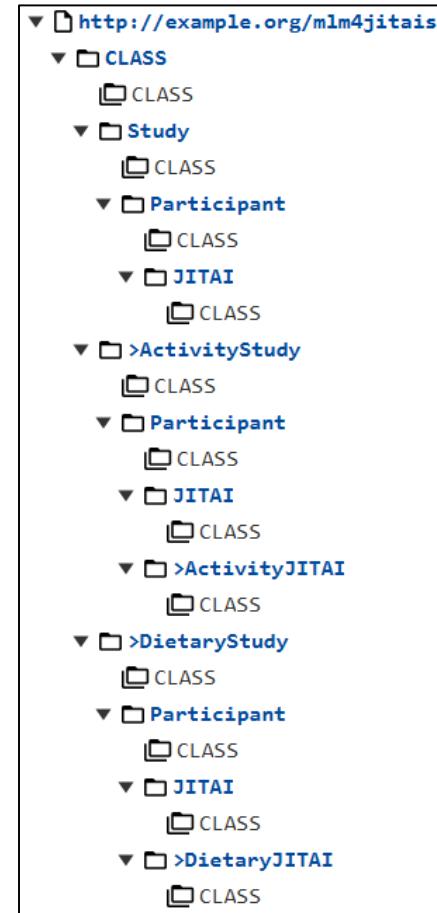


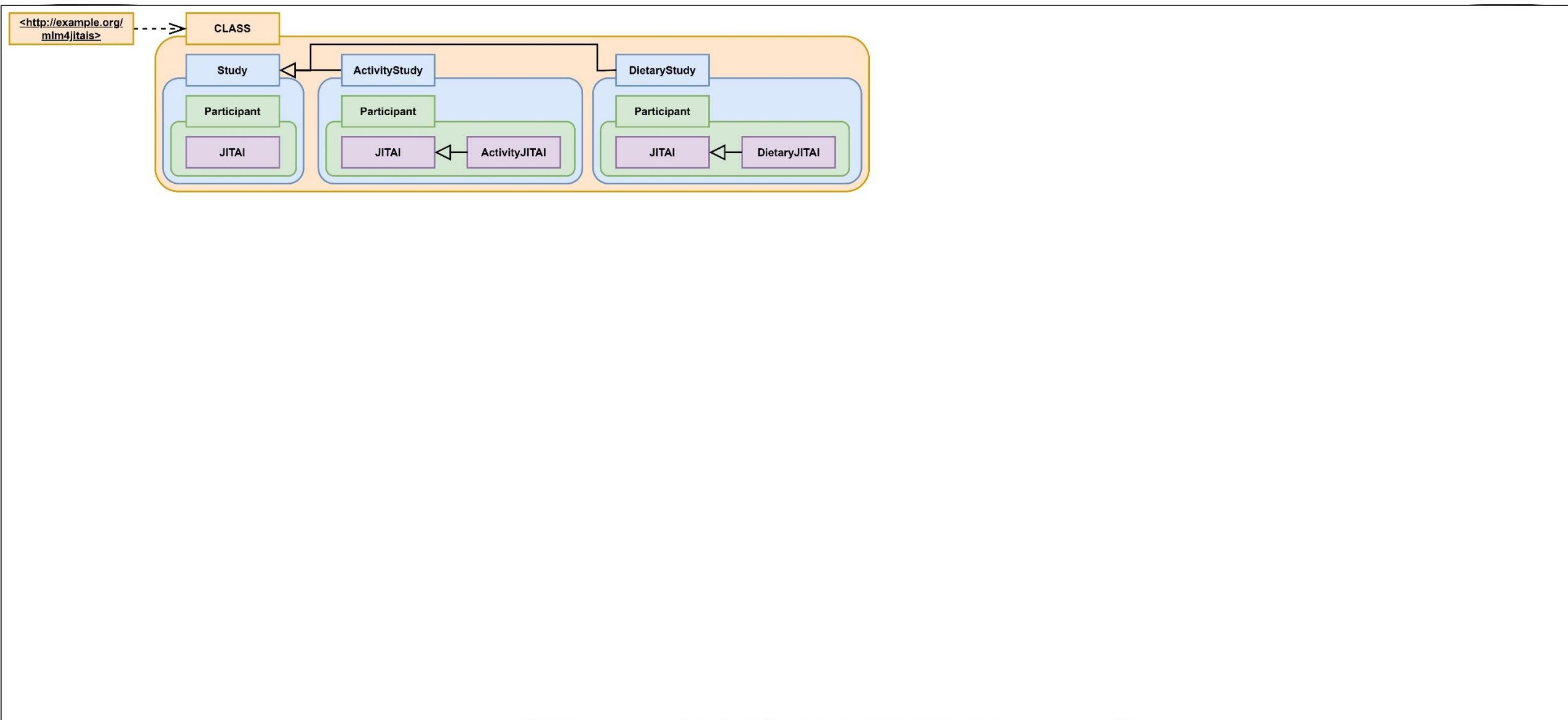


Task (4)

- Add class *ActivityJITAI* as specialization of class *JITAI* in context of class *ActivityStudy*
 - Note: Leafs of class *ActivityJITAI* are concrete classes
- Add class *DietaryJITAI* as specialization of class *JITAI* in context of class *DietaryStudy*
 - Note: Leafs of class *DietaryJITAI* are concrete classes

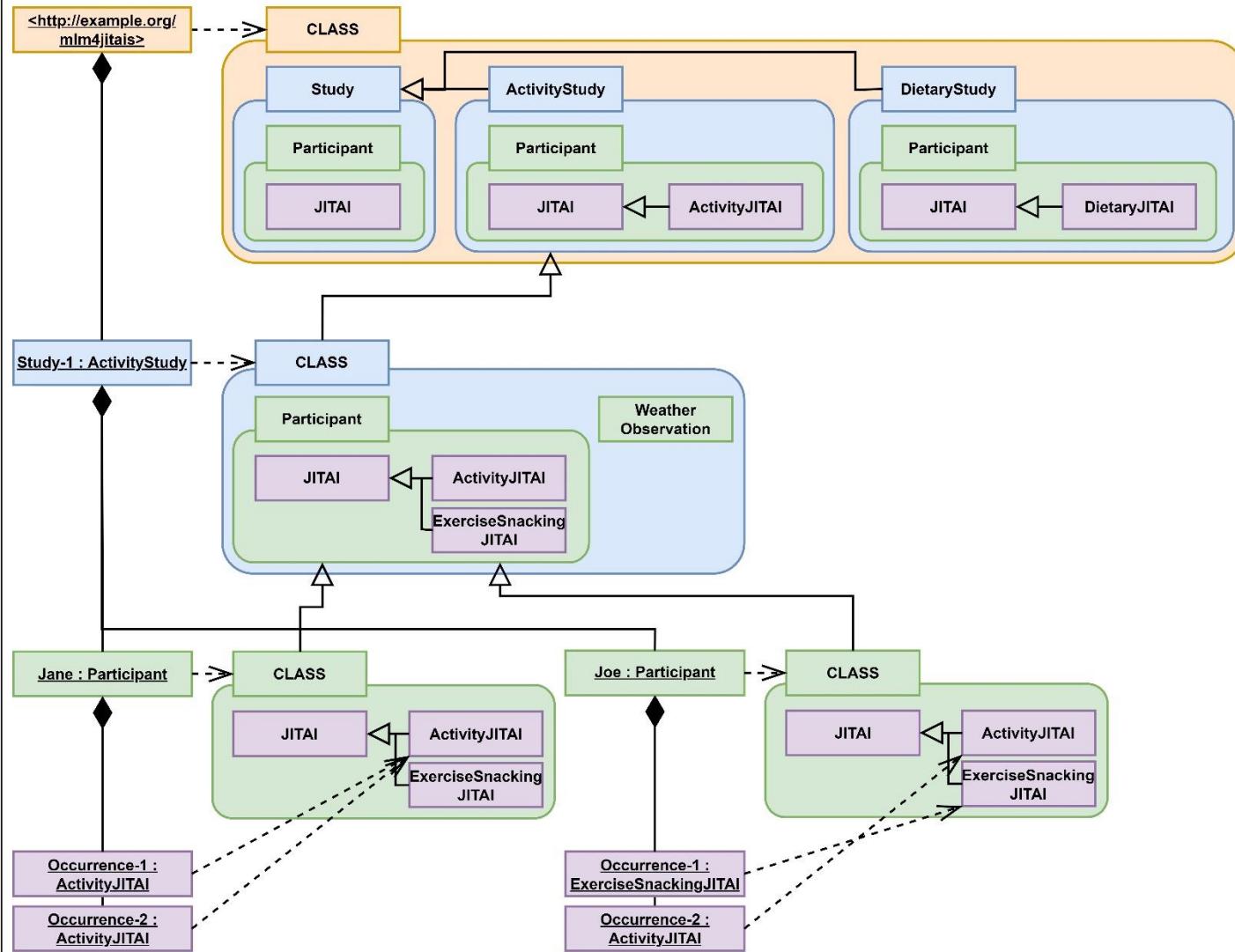
Expected outcome





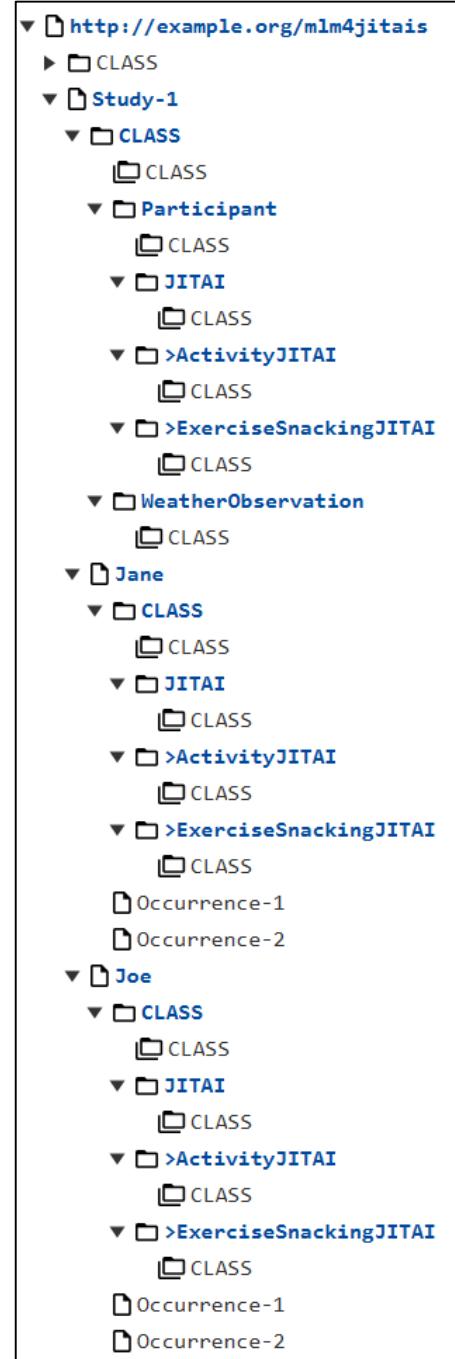
Task (5)

- Add *Study-1* as an instance of class *ActivityStudy*
 - Add class *WeatherObservation* for *Study-1*
 - Note: Individuals of class *WeatherObservation* are occurrences
 - Note: Leafs of class *WeatherObservation* are concrete classes
 - Add class *ExerciseSnackingJITAI* as specialization of class *JITAI* for *Study-1*
 - Note: Leafs of class *ExerciseSnackingJITAI* are concrete classes
 - Add *Jane* and *Joe* as instances of class *Participant* of *Study-1*
 - For *Jane* add the following occurrences:
 - *Occurrence-1* as an instance of class *ActivityJITAI* with a *suggestedDuration* of 6
 - *Occurrence-2* as an instance of class *ActivityJITAI* with a *suggestedDuration* of 23
 - For *Joe* add the following occurrences:
 - *Occurrence-1* as an instance of class *ExerciseSnackingJITAI* with a *suggestedDuration* of 2
 - *Occurrence-2* as an instance of class *ActivityJITAI* with a *suggestedDuration* of 33



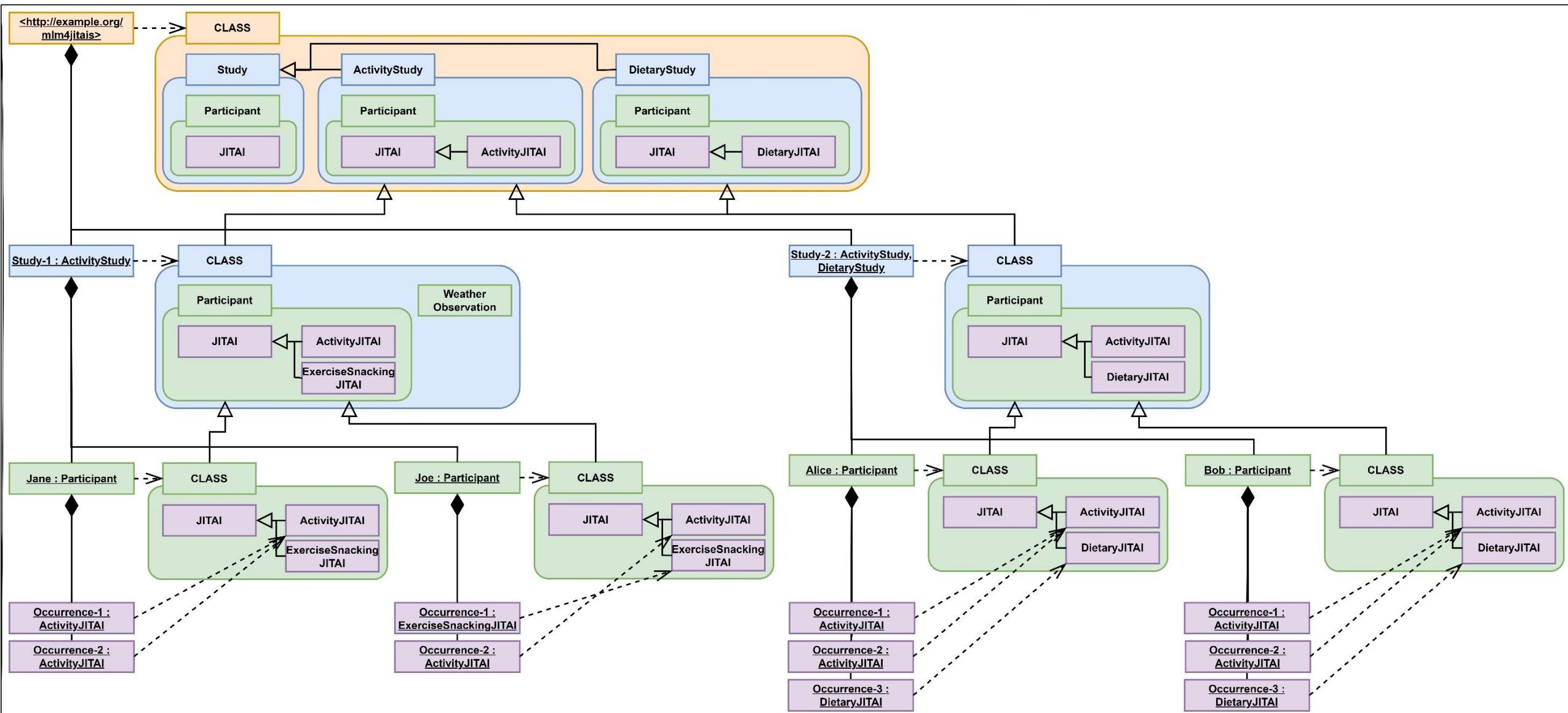
Task (5)

Expected outcome



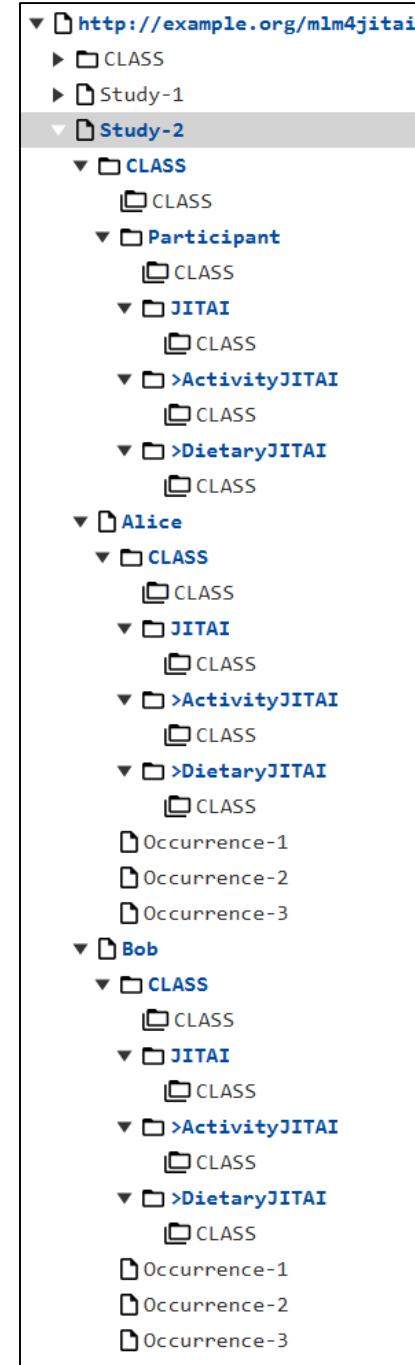
Task (6)

- Add *Study-2* as an instance of classes *ActivityStudy* and *DietaryStudy*
 - Add *Alice* and *Bob* as instances of class *Participant* of *Study-2*
 - For *Alice* add the following occurrences:
 - *Occurrence-1* as an instance of class *ActivityJITA1* with a *suggestedDuration* of 6
 - *Occurrence-2* as an instance of class *ActivityJITA1* with a *suggestedDuration* of 23
 - *Occurrence-3* as an instance of class *DietaryJITA1* with *message* "You are doing great!"
 - For *Bob* add the following occurrences:
 - *Occurrence-1* as an instance of class *ActivityJITA1* with a *suggestedDuration* of 2
 - *Occurrence-2* as an instance of class *ActivityJITA1* with a *suggestedDuration* of 33
 - *Occurrence-3* as an instance of class *DietaryJITA1* with *message* "You are doing bad!"



Task (6)

Expected outcome



Task (7)

- Define a SPARQL Query that counts the numbers of interventions for each instance of meta class
`<CLASS/ActivityStudy/Participant/ActivityJITAI/CLASS>`.
- Expected result

clabject	numberOfInterventions
<code><Study-2/Alice/CLASS/ActivityJITAI></code>	2
<code><Study-2/CLASS/Participant/ActivityJITAI></code>	4
<code><Study-1/CLASS/Participant/ActivityJITAI></code>	3
<code><Study-1/Jane/CLASS/ActivityJITAI></code>	2
<code><CLASS/ActivityStudy/Participant/ActivityJITAI/CLASS></code>	7
<code><Study-2/Bob/CLASS/ActivityJITAI></code>	2
<code><Study-1/Joe/CLASS/ActivityJITAI></code>	1

Task (8)

- Define a SPARQL Query that provides the average *suggestedDuration* for each instance of meta class <CLASS/Study/Participant/JITAI/CLASS>.
 - Expected result

Task (9)

- Define a SPARQL Query that selects all *ActivityJITAI* classes of individual participants using the navigation via n: properties only.
- Expected result

```
-----  
| clabject |  
=====|  
| <Study-1/Jane/CLASS/ActivityJITAI> |  
| <Study-1/Joe/CLASS/ActivityJITAI> |  
| <Study-2/Bob/CLASS/ActivityJITAI> |  
| <Study-2/Alice/CLASS/ActivityJITAI> |  
-----
```

Task (10)

- Define a SPARQL Query that selects the total number of JITAI occurrences for each study.
- Expected result

clabject	numberOfInterventions
<Study-2/CLASS/Participant/JITAI>	6
<Study-1/CLASS/Participant/JITAI>	4

Evaluation

- There is a voluntary and anonymized survey, you are kindly invited to participate after completing the tutorial and working through the task.
- Even if you could not successfully solve all subtasks, any feedback is highly important and appreciated.
- Here is the link (< 5min):
 - <https://survey.jku.at/jku2018/index.php/338561?lang=en>
- **Thank you!**