Course: Intelligent Systems

Unit 3: Ontology Engineering

Patterns in Knowledge Representation

Mari Carmen Suárez de Figueroa Baonza
Course 2022 – 2023
Technical University of Madrid

License

Index

What is a Pattern?

- The term **pattern** derives from Middle Latin "*patronus*"
  - Meaning "*patron*", and, metonymically, "*exemplar*", that is something proposed for imitation
- The term **design pattern** was introduced in the 1970's by Christopher Alexander for shared guidelines that help solve design problems
- In Software Engineering, a **design pattern** is defined as a simple and elegant solution to specific problems in object-oriented software design
  - A design pattern provides a common vocabulary for designers to communicate, document, and explore design alternatives
  - A design pattern is a generic description of how to solve a modelling problem that can be used in many different situations

Design Patterns in Knowledge Representation

- A **Pattern** is something proposed for imitation
- A **Design Pattern** in knowledge representation is a modelling solution to solve a recurrent and well-known representation problem
  - How to represent the following situations?
    - *"Mariano Fernández-López and Mari Carmen Suárez-Figueroa are senior researchers. Mariano is also associate professor and Mari Carmen is lecturer"*
    - *"Mariano and Mari Carmen co-participate at the ISWC 2019 conference"*
- **Design patterns**
  - facilitate the solution of modelling issues
  - improve interoperability through using well-proven solutions and best practices

Types of Design Patterns in Knowledge Representation

- There are **different types of design patterns**
  - Each type addresses different kinds of problems
  - Each type can be represented with different levels of formality
- **Logical Design Patterns** are formal expressions, whose only parts are expressions from a logical vocabulary (e.g., OWL DL), that solve a problem of expressivity
  - Logical patterns are independent from a specific domain of interest, i.e. they are content-independent
  - Logical patterns solve design problems where the primitives of the representation language do not directly support certain logical constructs
- **Content Design Patterns** encode conceptual, rather than logical design patterns
  - Content patterns propose patterns for solving design problems for the domain problem
  - Content patterns address content problems

Index

Relations in Knowledge Representation

- **Relations** are a key concept in structured knowledge representations
  - Examples include temporal relations, spatial relations, family relations, social relations, administrative organizations, military hierarchies, etc.
- The **arity of a relation** is the number of entities that it associates
  - Unary or Monadic represents properties for entities
  - Binary or dyadique associates two entities
  - Ternary or triadique → N-ary

Relations in Knowledge Representation

- The most common type of relation is **binary relation**
  - *The King Felipe VI is married with the Queen Letizia*
  - *Luka Modrić belongs to Real Madrid*
  - *Asunción Gómez-Pérez is author of the Ontological Engineering book*
- Sometimes, **relationships that hold between one subject and two or more objects** should be represented
  - *Jack has given the TV series 'Game of Thrones' a rating of 8*
  - *Asunción Gómez-Pérez collaborated with Mariano Fernández-López on the "Searching for a time ontology" paper*
  - *Vicente Martínez is the coordinator of the subject 'Artificial Intelligence'*
- Relations that link an individual to more than one individual or value are called **N-ary relations**

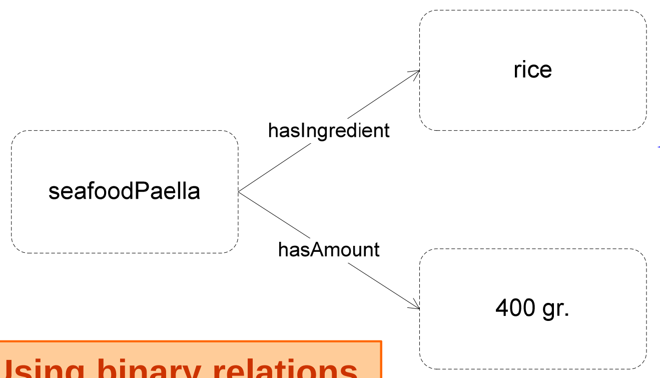
N-ary Relations: Examples

- A **relation** initially thought to be binary, **needs a further argument**
  - *Christine has breast tumor with high probability*
- **Two binary properties turn out to always go together** and should be represented as one n-ary relation
  - *Peter has temperature, which is high, but falling*
- From the beginning **the relation is really amongst several things**
  - *María buys the “Design Pattern” book in Amazon for 45 euros as a birthday gift*
- One or more of the arguments is fundamentally **a sequence rather than a single individual**
  - *United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK*

http://www.w3.org/TR/swbp-n-aryRelations/

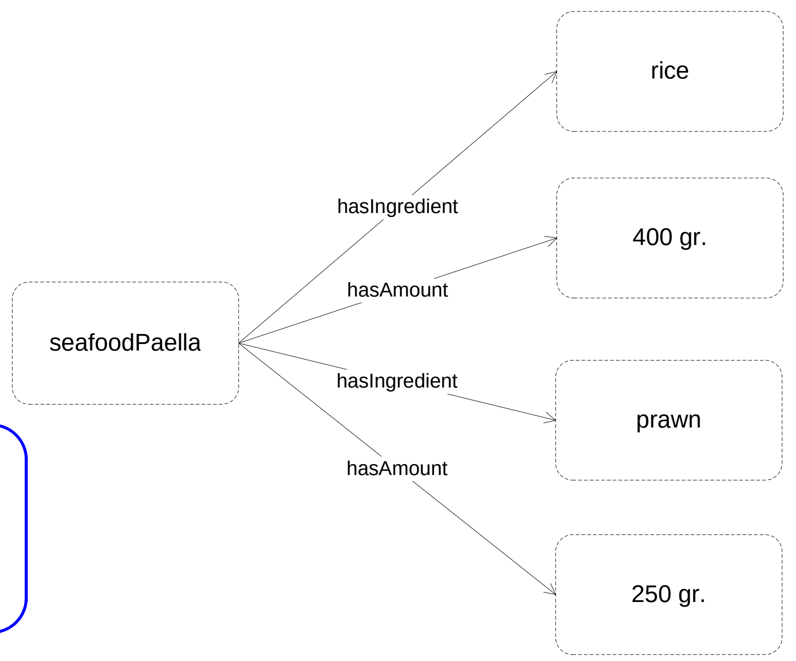
How to represent the following situation? (a)

- *Seafood paella has rice as ingredient and the amount should be 400 gr.*



What are the limitations with this solution?
What happen when other ingredients (such as prawns) are included in the model?

Using binary relations

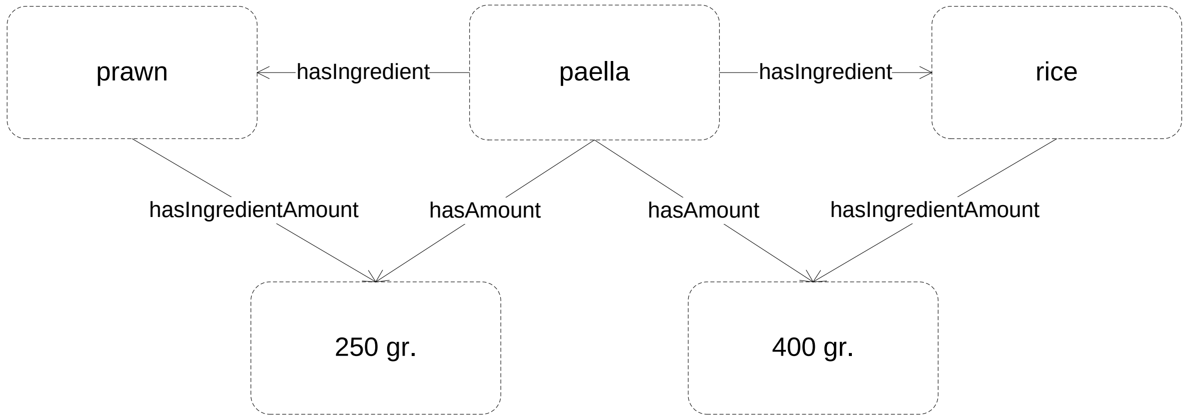


What are the limitations with this solution?
How do we know the amount of each ingredient?

How to represent the following situation? (a)

- *Seafood paella has as ingredients (a) 400 gr of rice and (b) 250 gr of prawns*

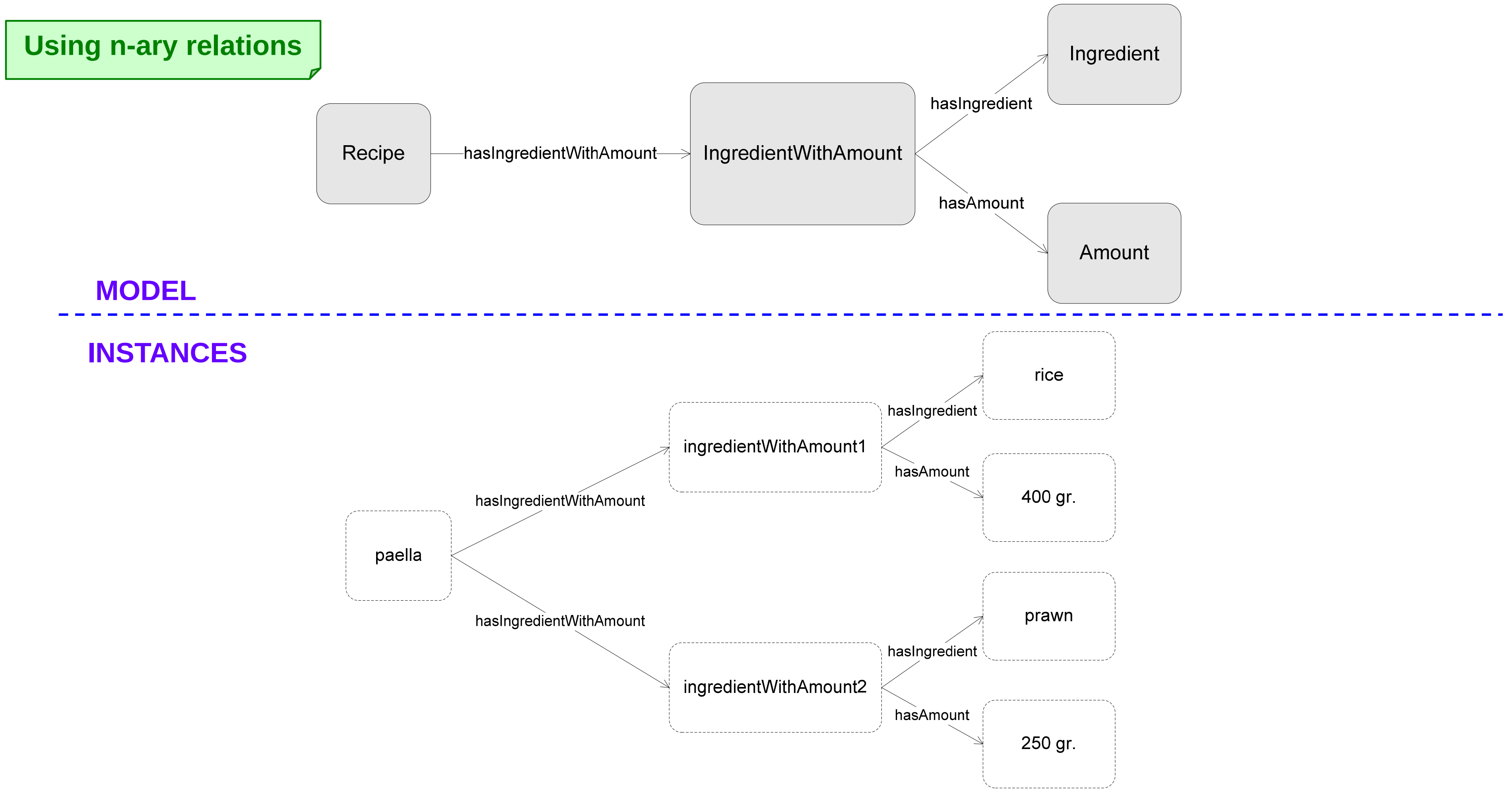
Using binary relations



- **Conclusion**: An n-ary relation cannot be split up into ‘n’ binary relations, because the relations it defines are all interconnected in some way

How to represent the following situation? (a)

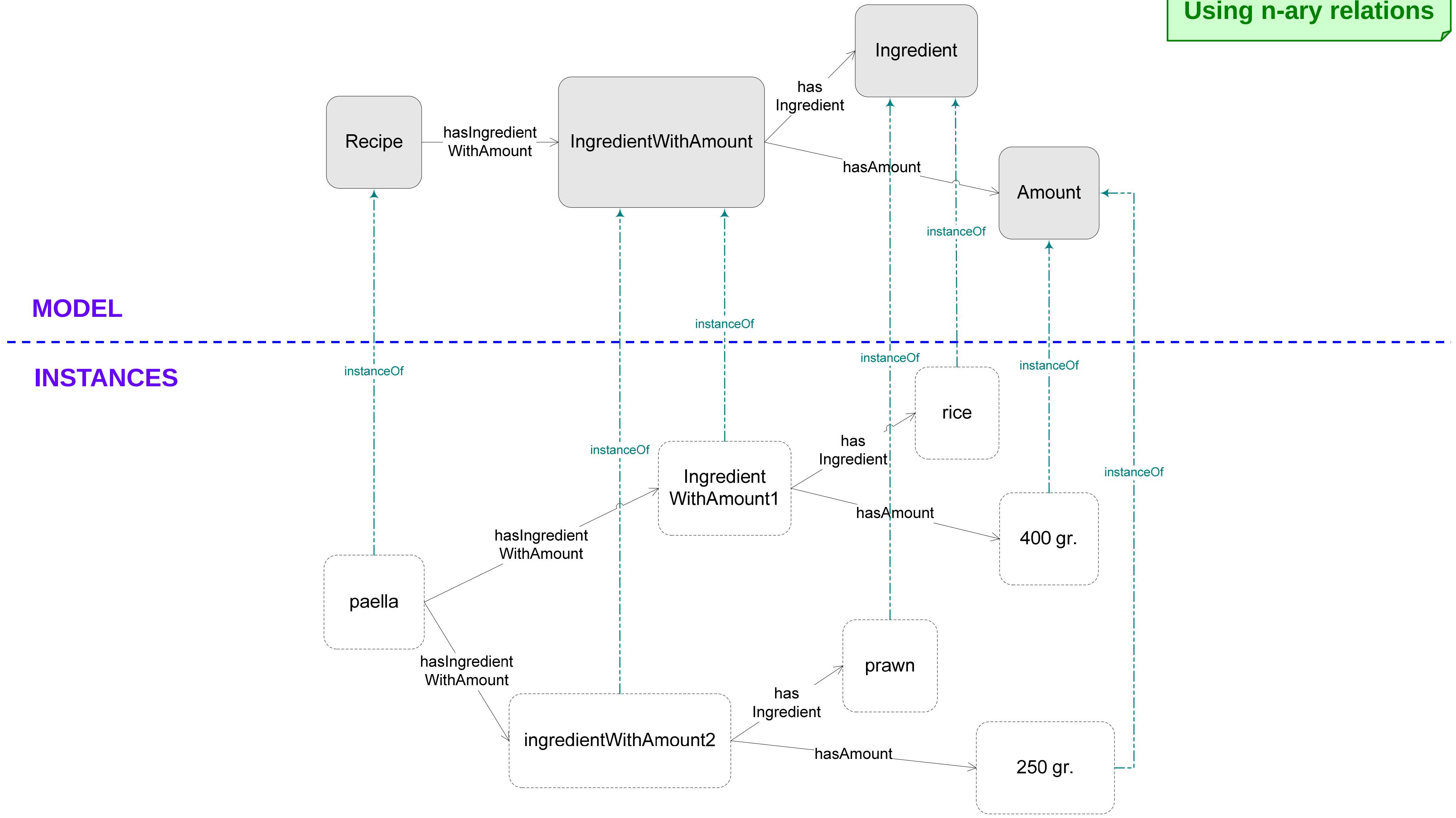
- **(General) Situation**: A cook recipe has different ingredients with particular amounts
- **Functional Representation:**
  - ***IngredientWithAmount (Recipe, Ingredient, Amount)***



How to represent the following situation? (a)

- ***IngredientWithAmount (Recipe, Ingredient, Amount)***

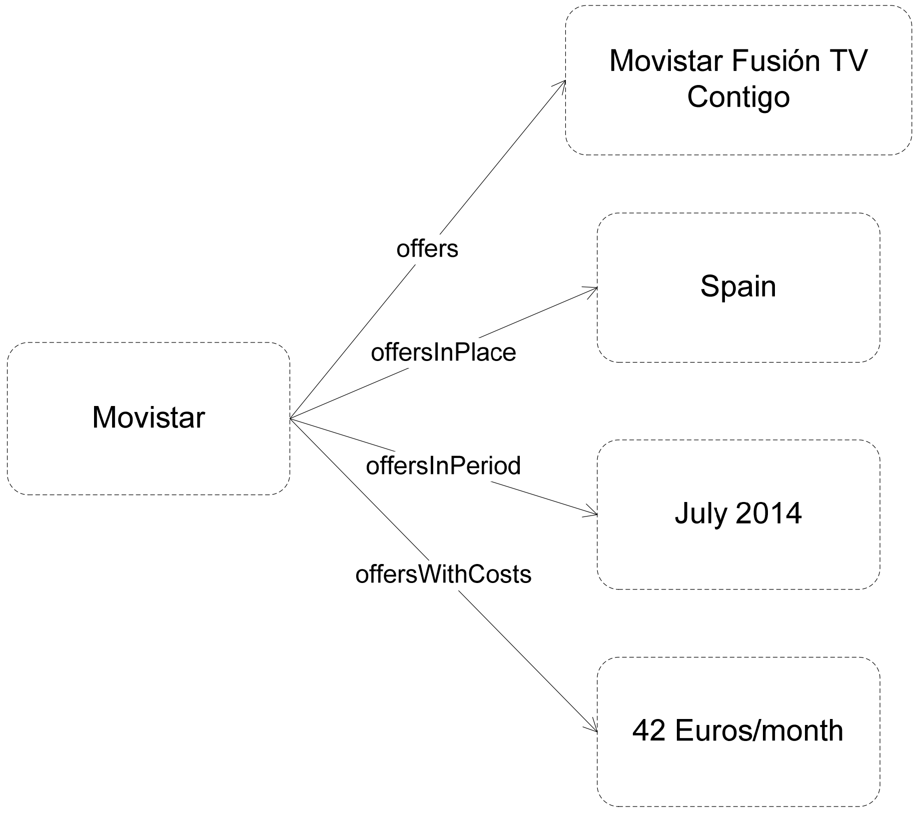
Using n-ary relations



How to represent the following situation? (b)

- **(Specific) Situation**: *Movistar offers "Movistar Fusión TV Contigo" in Spain during July 2014 for 42 euros/month*

Using binary relations



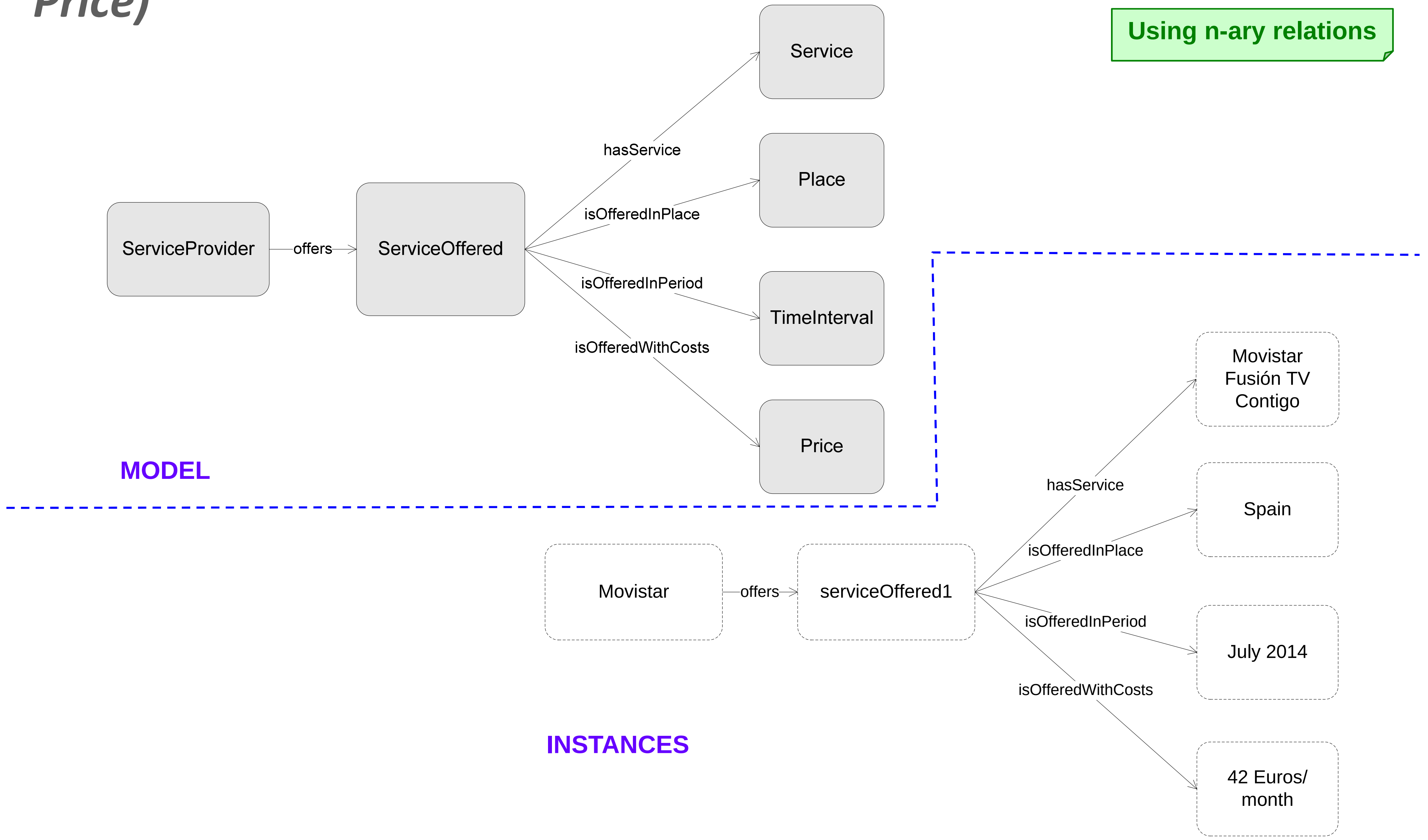
What are the limitations with this solution?
What happen when Movistar provides other services?

How to represent the following situation? (b)

- **(Specific) Situation**: *Movistar offers "Movistar Fusión TV Contigo" in Spain during July 2014 for 42 euros/month*
- **(General) Situation**: A service provider offers a service at a place in a given period of time with a particular price
- **Functional Representation:**
  - ***ServiceOffered (ServiceProvider, Service, Place, TimeInterval, Price)***

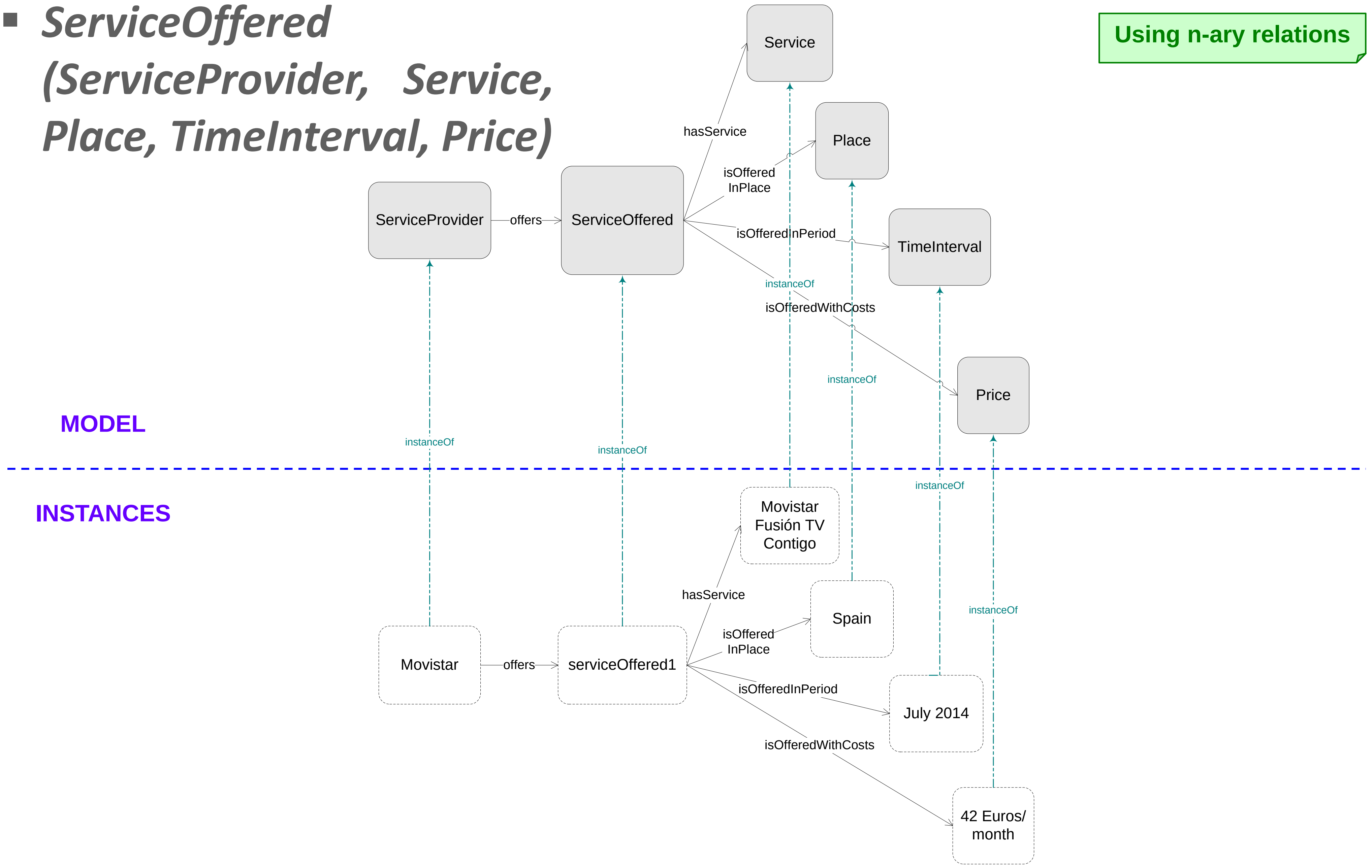
How to represent the following situation? (b)

- ***ServiceOffered (ServiceProvider, Service, Place, TimeInterval, Price)***



How to represent the following situation? (b)

- ***ServiceOffered (ServiceProvider, Service, Place, TimeInterval, Price)***

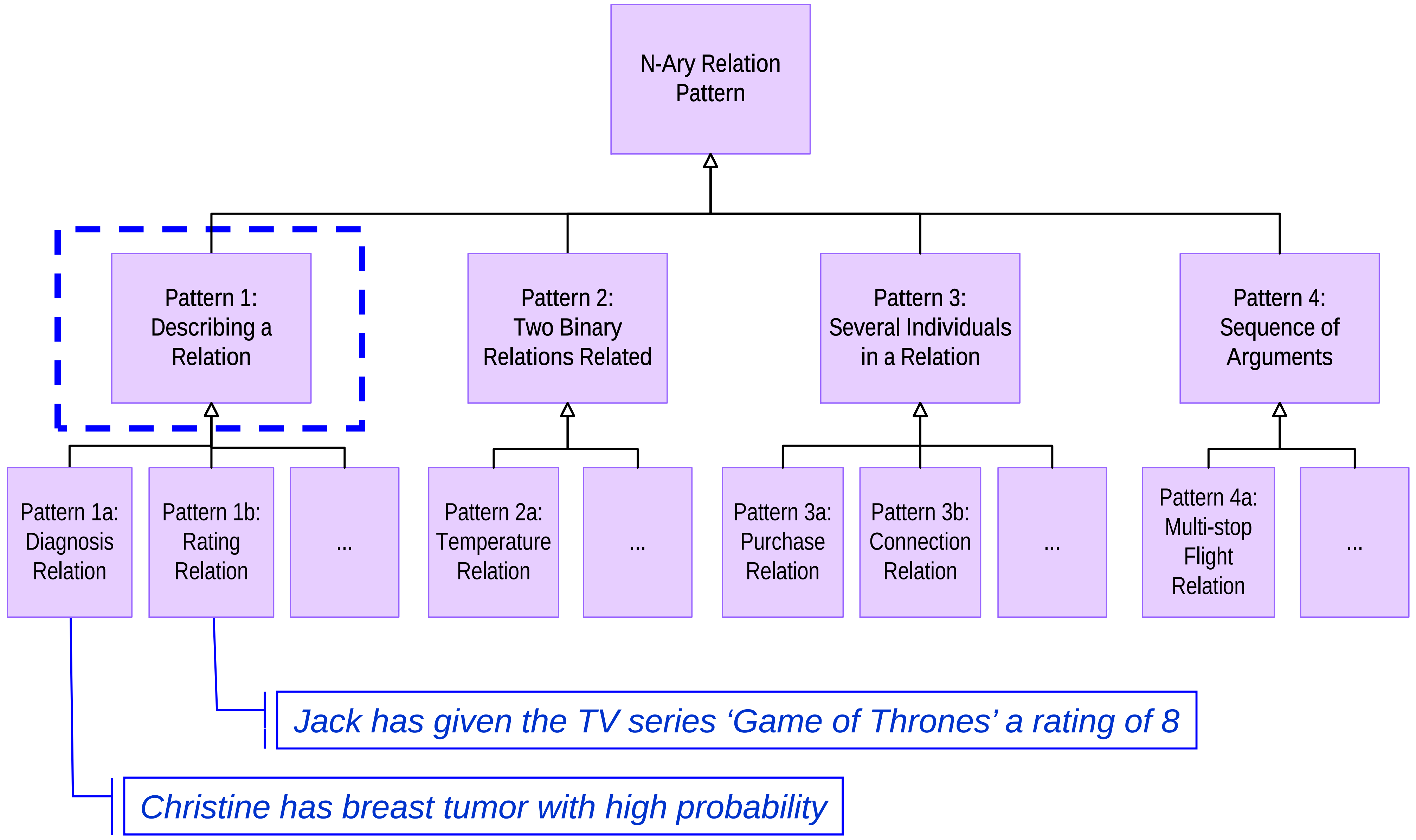


N-ary Relation Pattern (I)

- There are a lot of situations in which the natural and convenient way to represent certain concepts is to use relations to link an individual to more than just one individual or value (*n-ary relations*)
- This is a recurrent and well-known representation problem that is solved with the **N-ary Relation Pattern**
  - The idea is to represent the n-ary relation as a **class** rather than a property
    - To create a new class and new properties to represent the n-ary relation
    - Additional properties provide binary links to each argument of the relation
    - An instance of the relation linking the ‘n’ individuals is then an instance of the new class

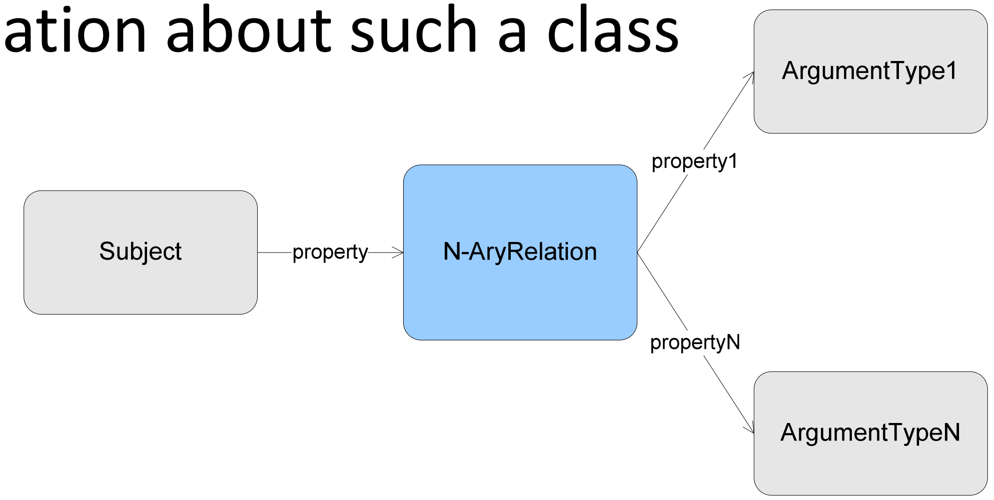
Index

Specializations of the N-ary Relation Pattern


N-Ary Relation Pattern
Pattern 1: Describing a Relation
Pattern 2: Two Binary Relations Related
Pattern 3: Several Individuals in a Relation
Pattern 4: Sequence of Arguments
Pattern 1a: Diagnosis Relation
Pattern 1b: Rating Relation
...
Pattern 2a: Temperature Relation
...
Pattern 3a: Purchase Relation
Pattern 3b: Connection Relation
...
Pattern 4a: Multi-stop Flight Relation
...
Jack has given the TV series 'Game of Thrones' a rating of 8
Christine has breast tumor with high probability

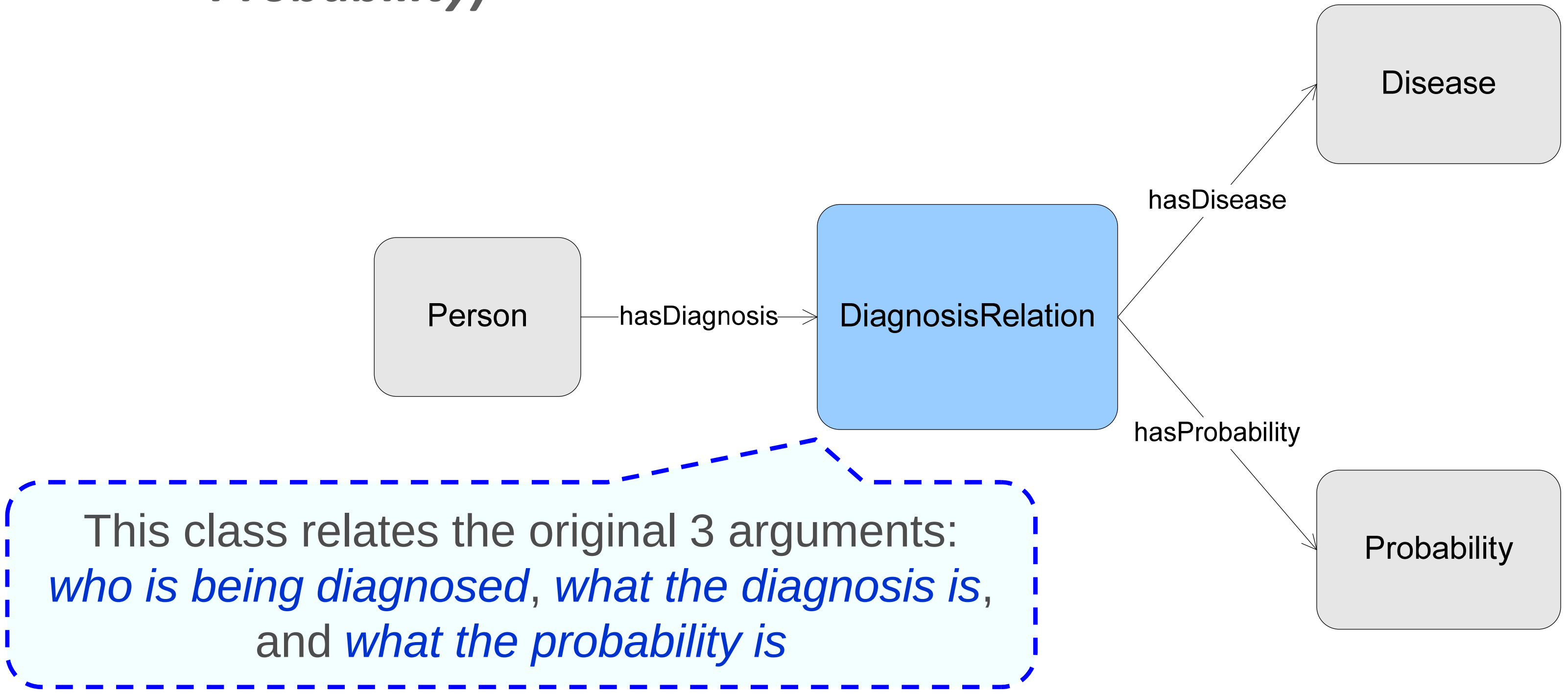
Pattern 1: Describing a Relation

- **A relation needs additional information**: an additional attribute describing a relation
  - *Christine has breast tumor with high probability*
  - *Jack has given the TV series 'Game of Thrones' a rating of 8*
- **Solution**: to create a class that represents the relation itself
  - with links from the subject of the relation to this class, and
  - with links from this class to all participants that represent additional information about such a class



Pattern 1: Describing a Relation. Pattern 1a: Diagnosis Relation

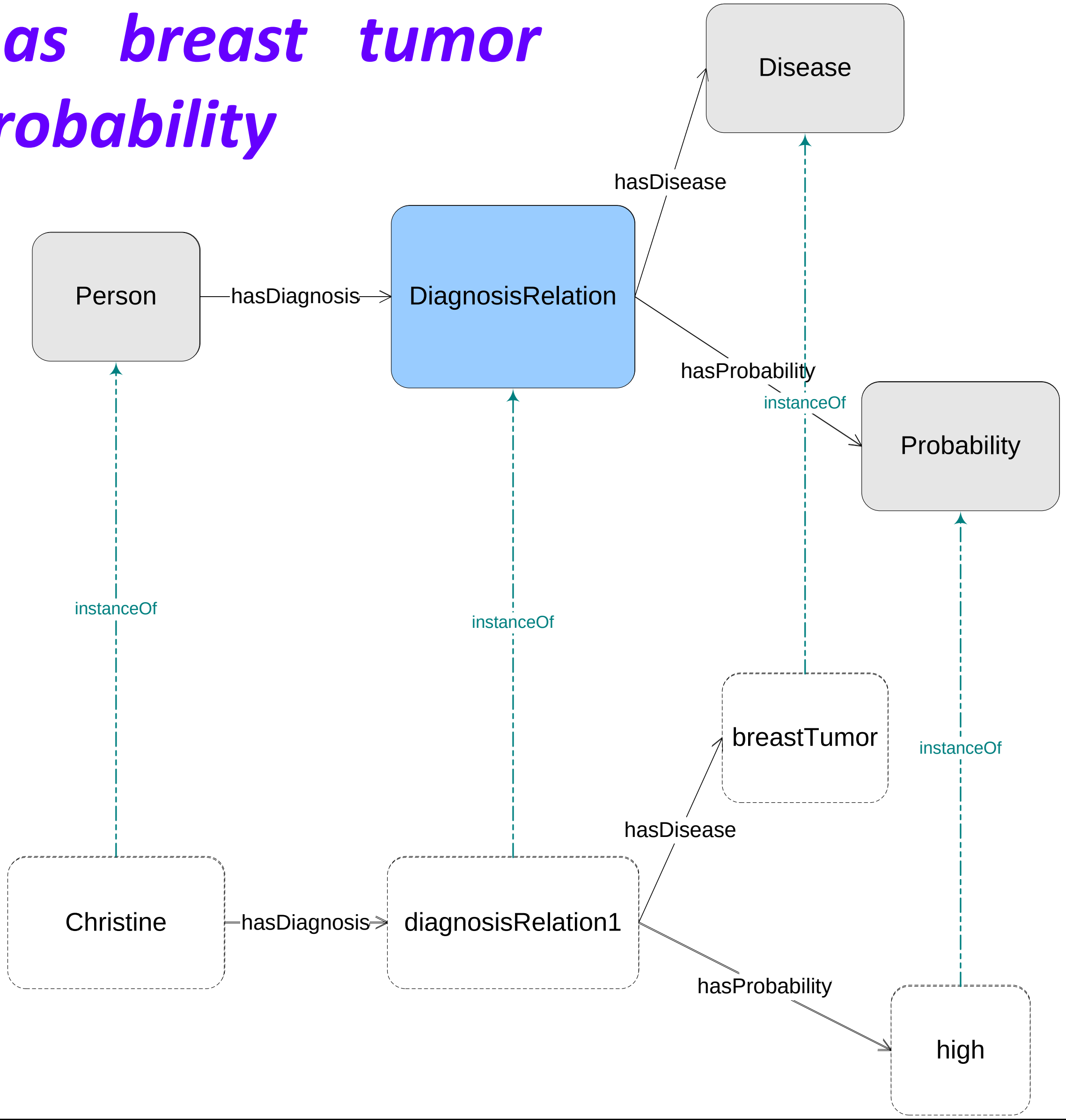
- **(General) Situation: A person has been diagnosed with a disease with a probability**
  - **(Specific) Situation**: *Christine has breast tumor with high probability*
    - **Functional Representation: *DiagnosisRelation (Person, Disease, Probability)***



This class relates the original 3 arguments: *who is being diagnosed*, *what the diagnosis is*, and *what the probability is*

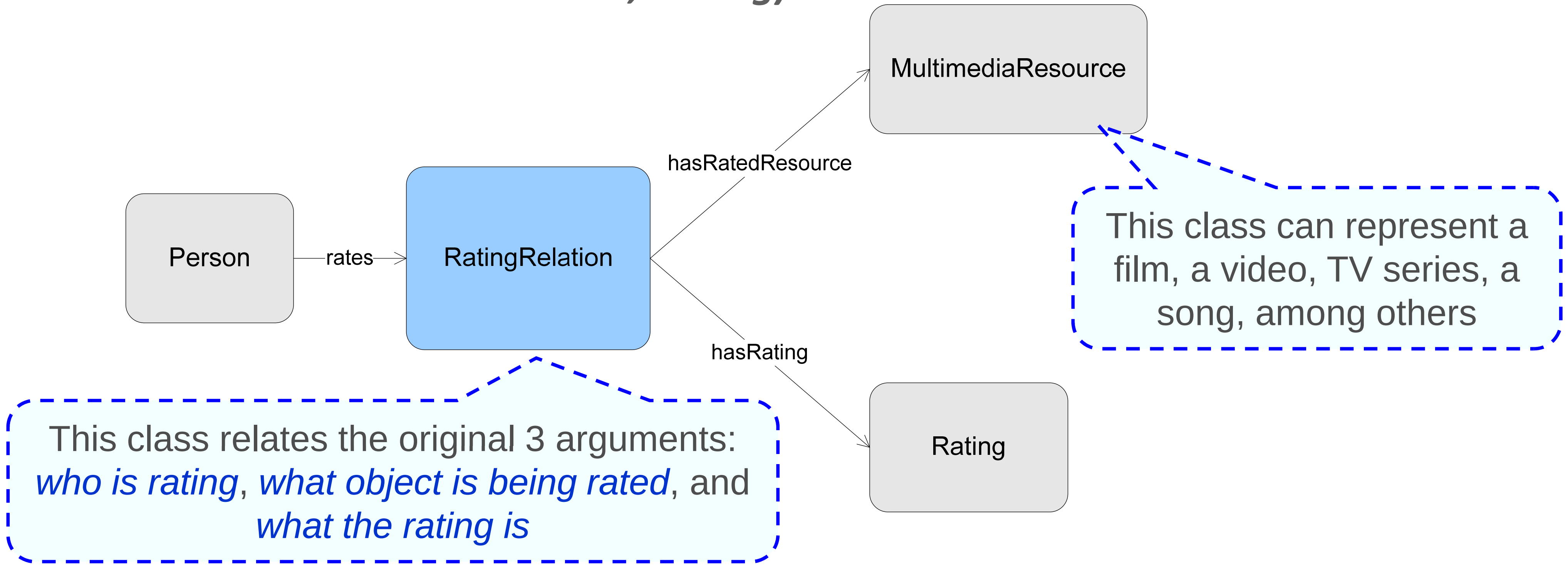
Pattern 1: Describing a Relation. Pattern 1a: Diagnosis Relation

- ***Christine has breast tumor with high probability***



Pattern 1: Describing a Relation. Pattern 1b: Rating Relation

- **(General) Situation: Someone has evaluated a multimedia resource with a rating**
  - **(Specific) Situation**: *Jack has given the TV series 'Game of Thrones' a rating of 8*
    - **Functional Representation:** ***RatingRelation (Person, MultimediaResource, Rating)***

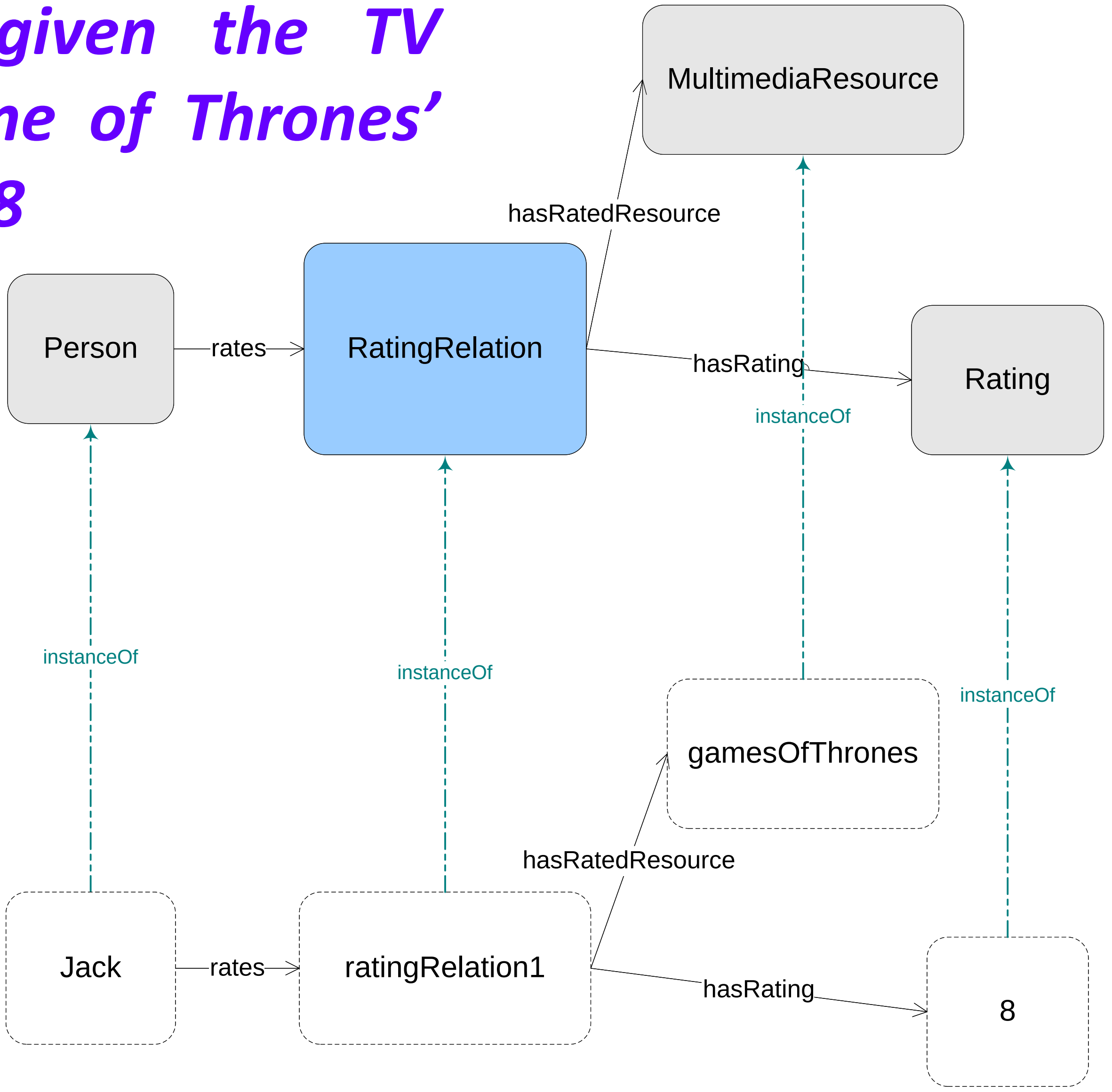


This class can represent a film, a video, TV series, a song, among others

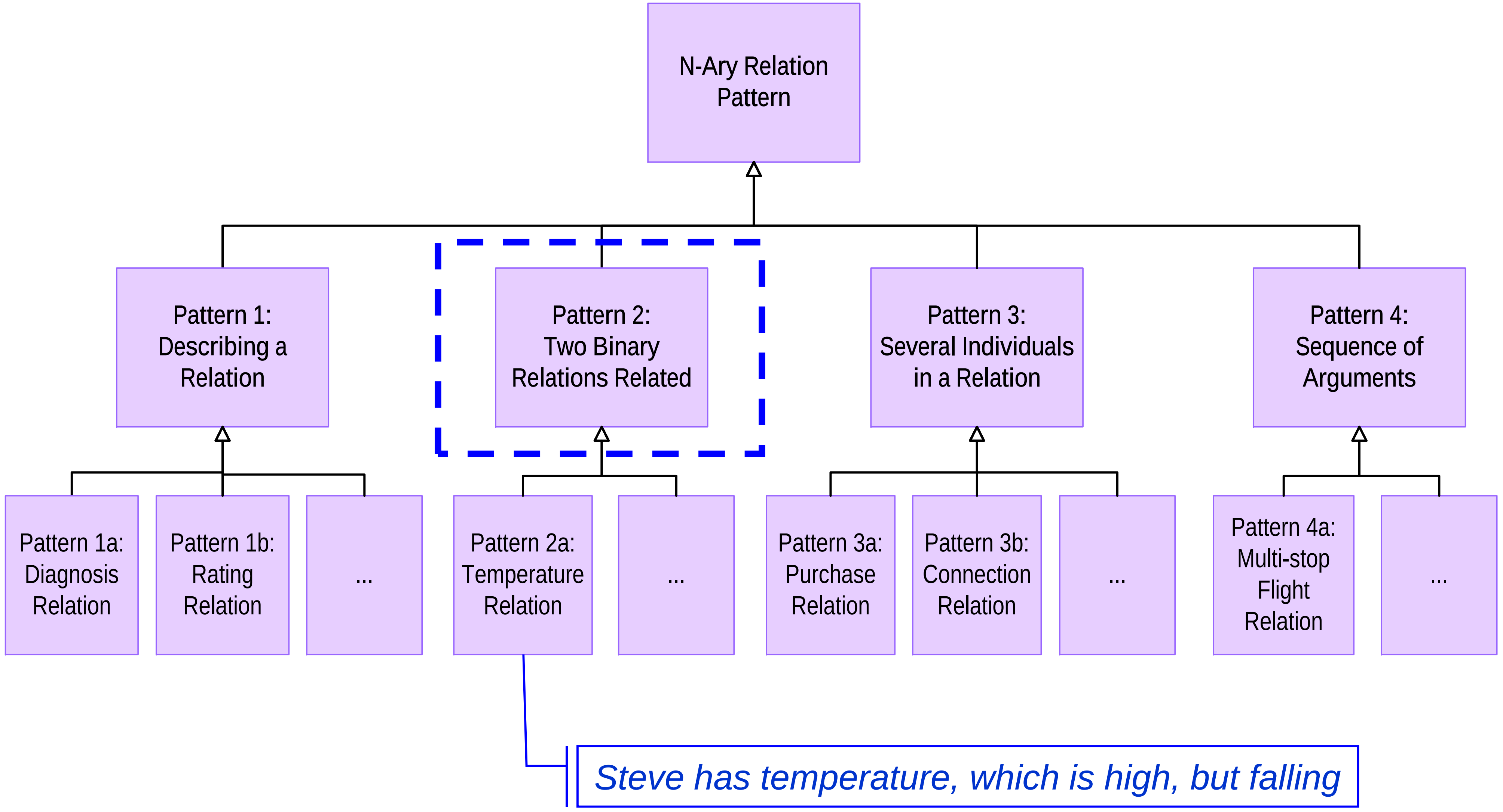
This class relates the original 3 arguments: *who is rating*, *what object is being rated*, and *what the rating is*

Pattern 1: Describing a Relation. Pattern 1b: Rating Relation

- ***Jack has given the TV series 'Game of Thrones' a rating of 8***

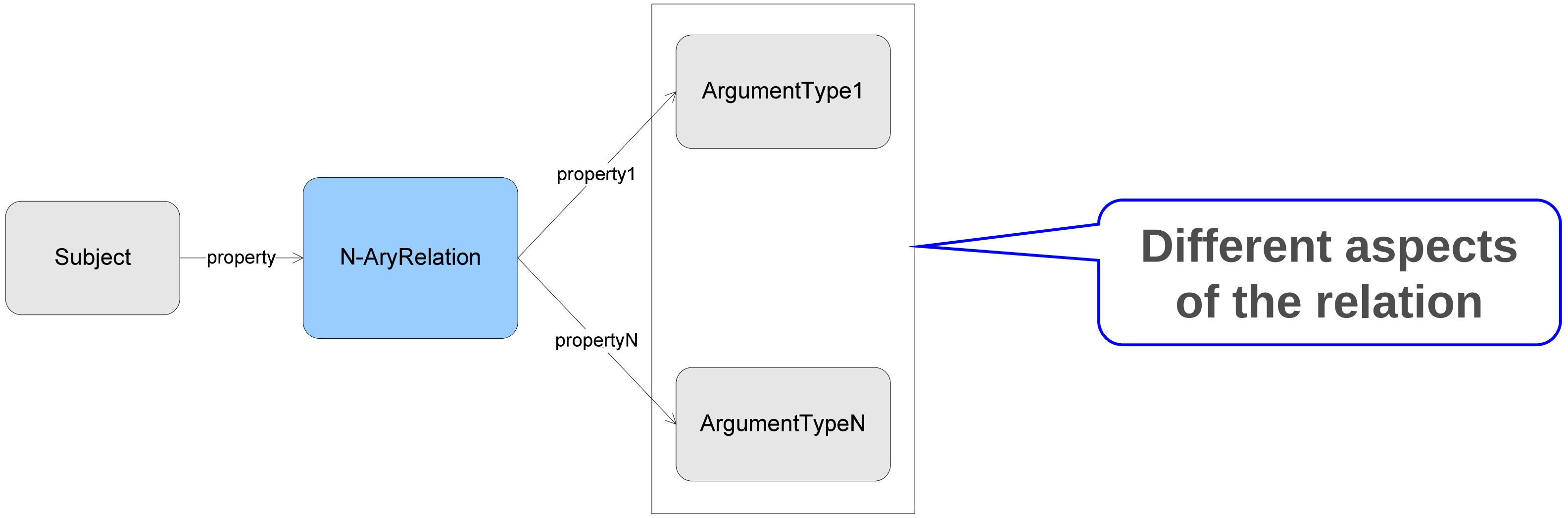


Specializations of the N-ary Relation Pattern


N-Ary Relation Pattern
Pattern 1: Describing a Relation
Pattern 2: Two Binary Relations Related
Pattern 3: Several Individuals in a Relation
Pattern 4: Sequence of Arguments
Pattern 1a: Diagnosis Relation
Pattern 1b: Rating Relation
...
Pattern 2a: Temperature Relation
...
Pattern 3a: Purchase Relation
Pattern 3b: Connection Relation
...
Pattern 4a: Multi-stop Flight Relation
...
Steve has temperature, which is high, but falling

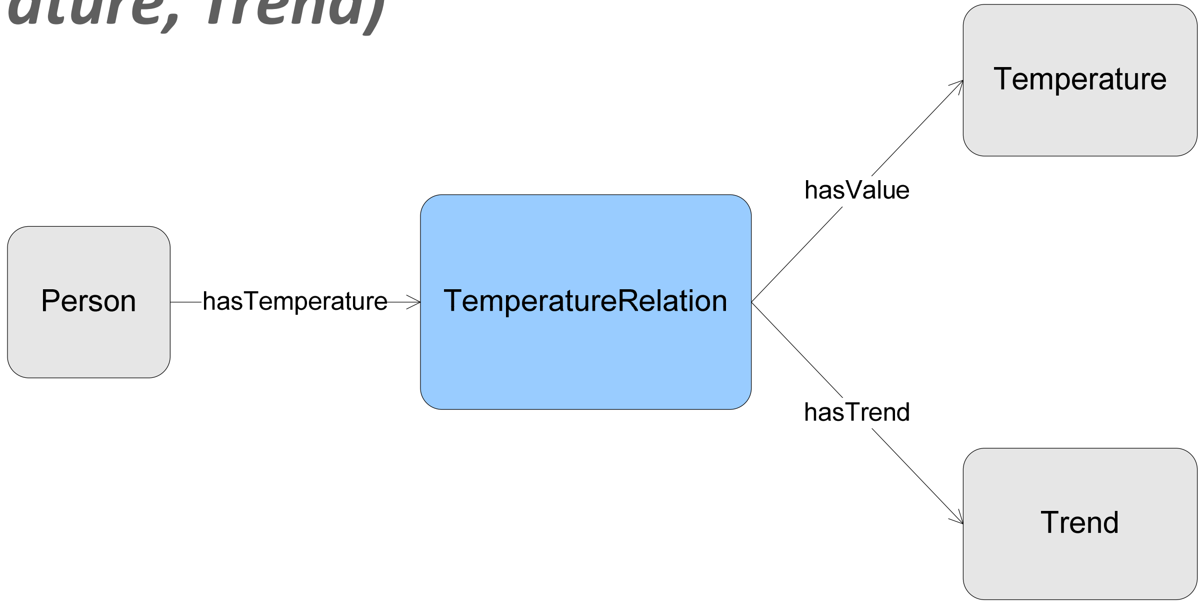
Pattern 2: Two Binary Relations Related

- **Two binary relations are related to each other**: different aspects of the same relation
  - *Steve has temperature, which is high, but falling*
    - Need: to represent different aspects of the temperature that Steve has
- **Solution**: to create a class that represents the relation between the subject and the complex object representing different facts about the specific relation



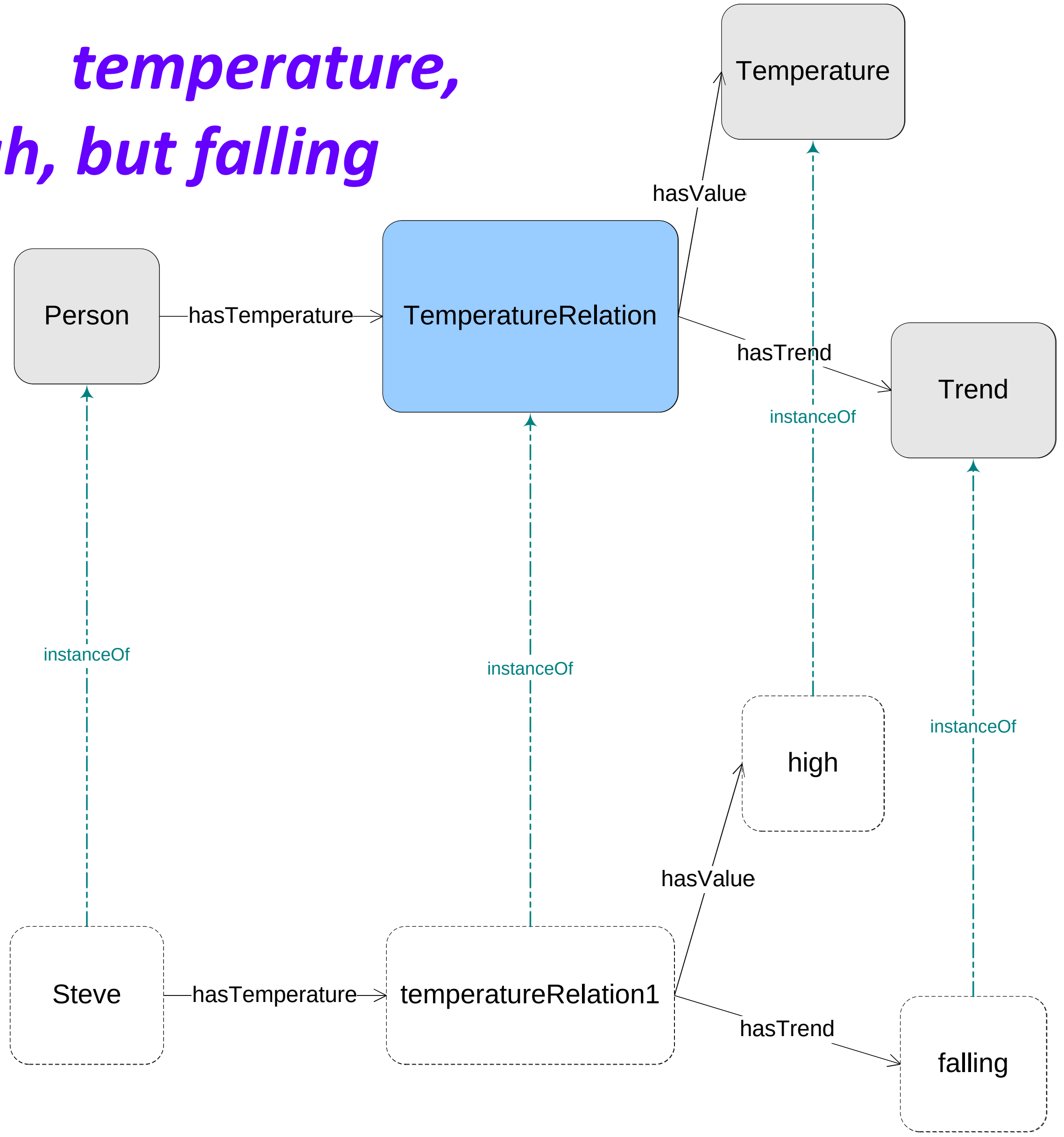
Pattern 2: Two Binary Relations Related. Pattern 2a: Temperature Relation

- **(General) Situation: Someone has a particular temperature, which can rise or fall**
  - **(Specific) Situation**: *Steve has temperature, which is high, but falling*
    - **Functional Representation: *TemperatureRelation (Person, Temperature, Trend)***

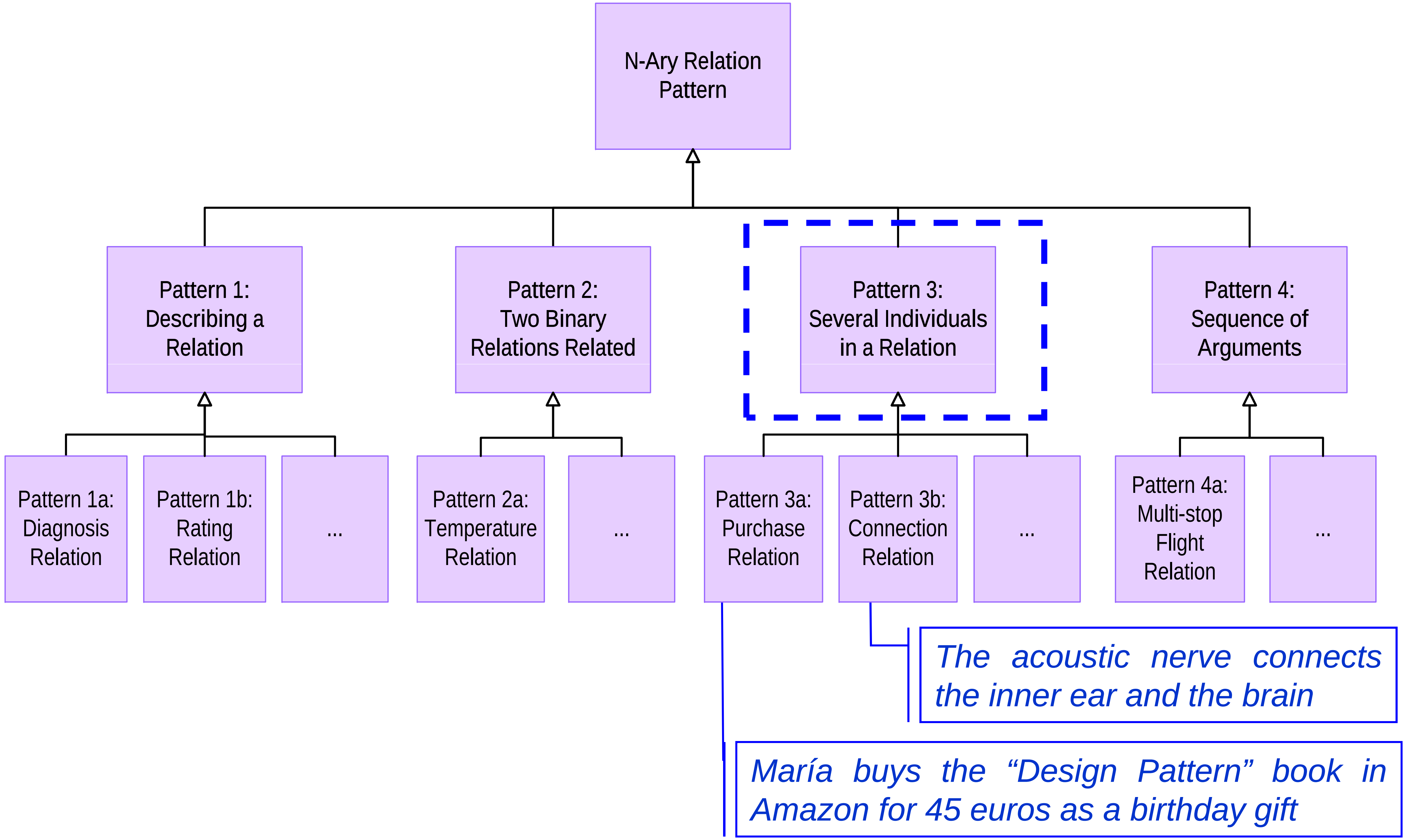


Pattern 2: Two Binary Relations Related. Pattern 2a: Temperature Relation

- ***Steve has temperature, which is high, but falling***

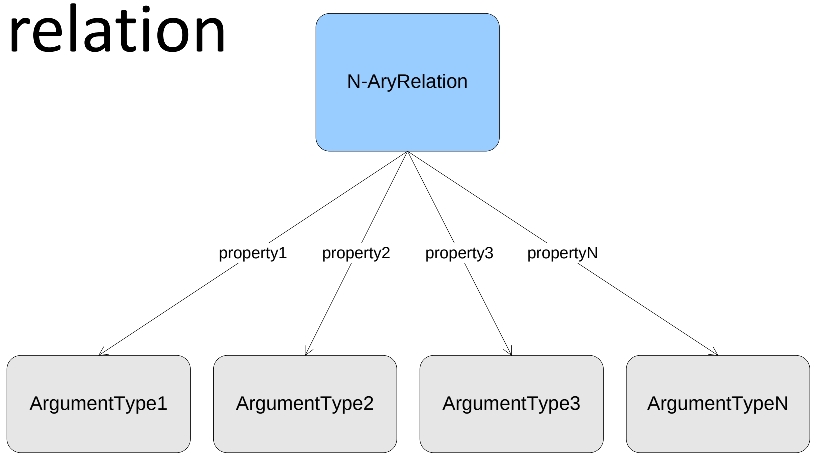


Specializations of the N-ary Relation Pattern


N-Ary Relation Pattern
Pattern 1: Describing a Relation
Pattern 2: Two Binary Relations Related
Pattern 3: Several Individuals in a Relation
Pattern 4: Sequence of Arguments
Pattern 1a: Diagnosis Relation
Pattern 1b: Rating Relation
...
Pattern 2a: Temperature Relation
...
Pattern 3a: Purchase Relation
Pattern 3b: Connection Relation
...
Pattern 4a: Multi-stop Flight Relation
...
The acoustic nerve connects the inner ear and the brain
María buys the “Design Pattern” book in Amazon for 45 euros as a birthday gift

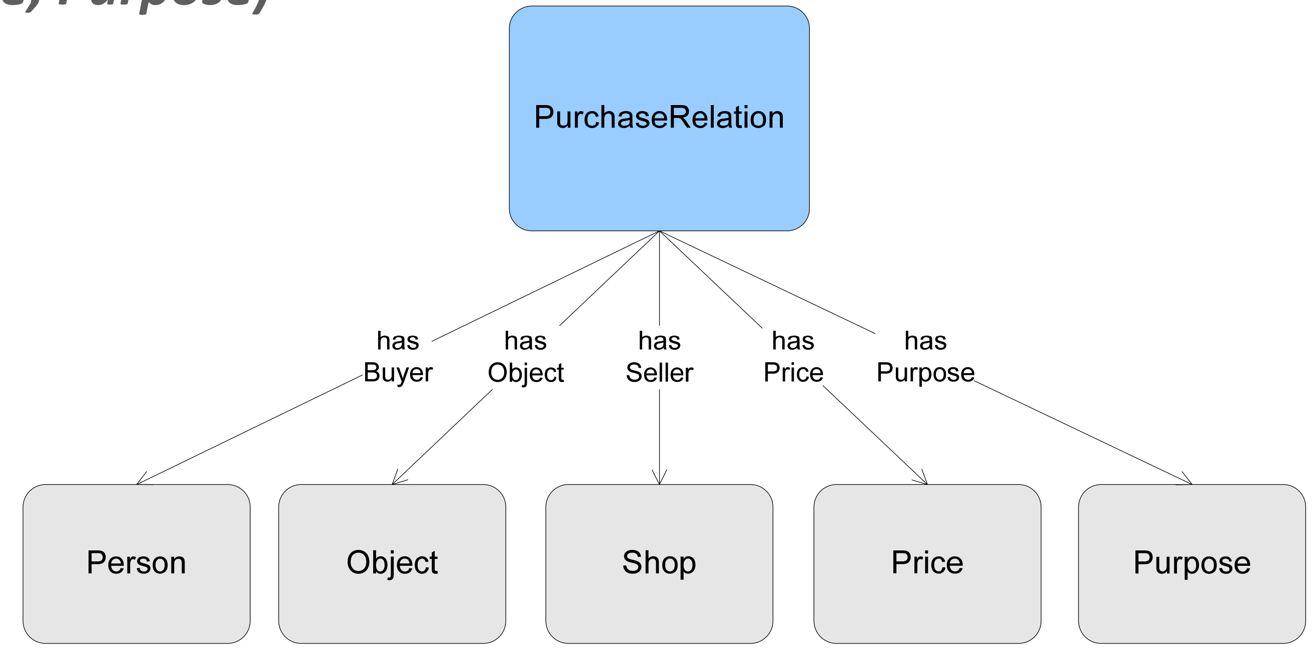
Pattern 3: Several Individuals in a Relation

- **A relation between several individuals**: N-ary relation with no distinguished participant
  - *María buys the "Design Pattern" book in Amazon for 45 euros as a birthday gift*
- In some cases the n-ary relationship links individuals that play different roles in a structure without any single individual standing out as the "owner" of the relation
- **Solution**: to create a class that represents the relation with links to all participants in the relation



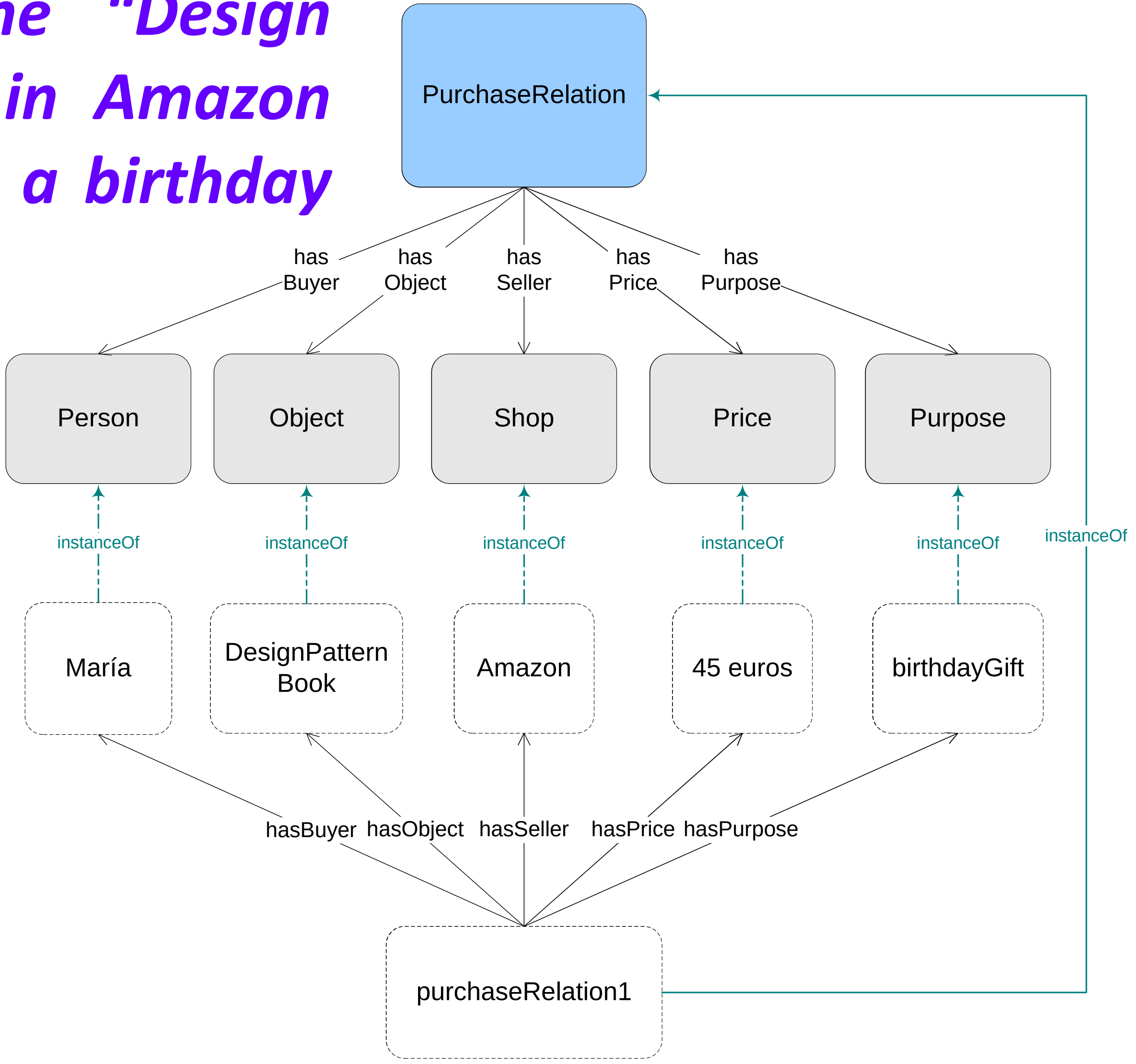
Pattern 3: Several Individuals in a Relation. Pattern 3a: Purchase Relation

- **(General) Situation: Someone buys an object in a shop for a price with a purpose**
  - **(Specific) Situation**: *María buys the "Design Pattern" book in Amazon for 45 euros as a birthday gift*
    - **Functional Representation: *PurchaseRelation (Person, Object, Shop, Price, Purpose)***



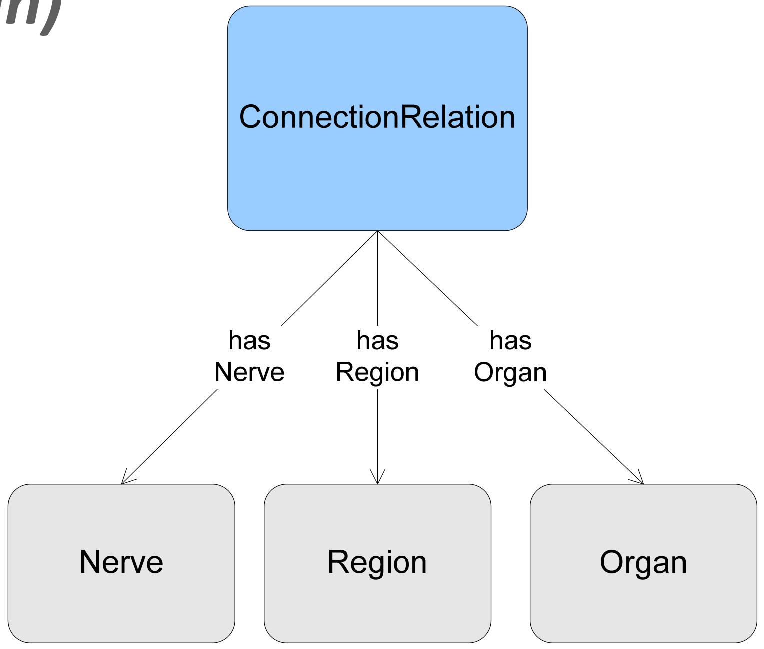
Pattern 3: Several Individuals in a Relation. Pattern 3a: Purchase Relation

- ***María buys the "Design Pattern" book in Amazon for 45 euros as a birthday gift***



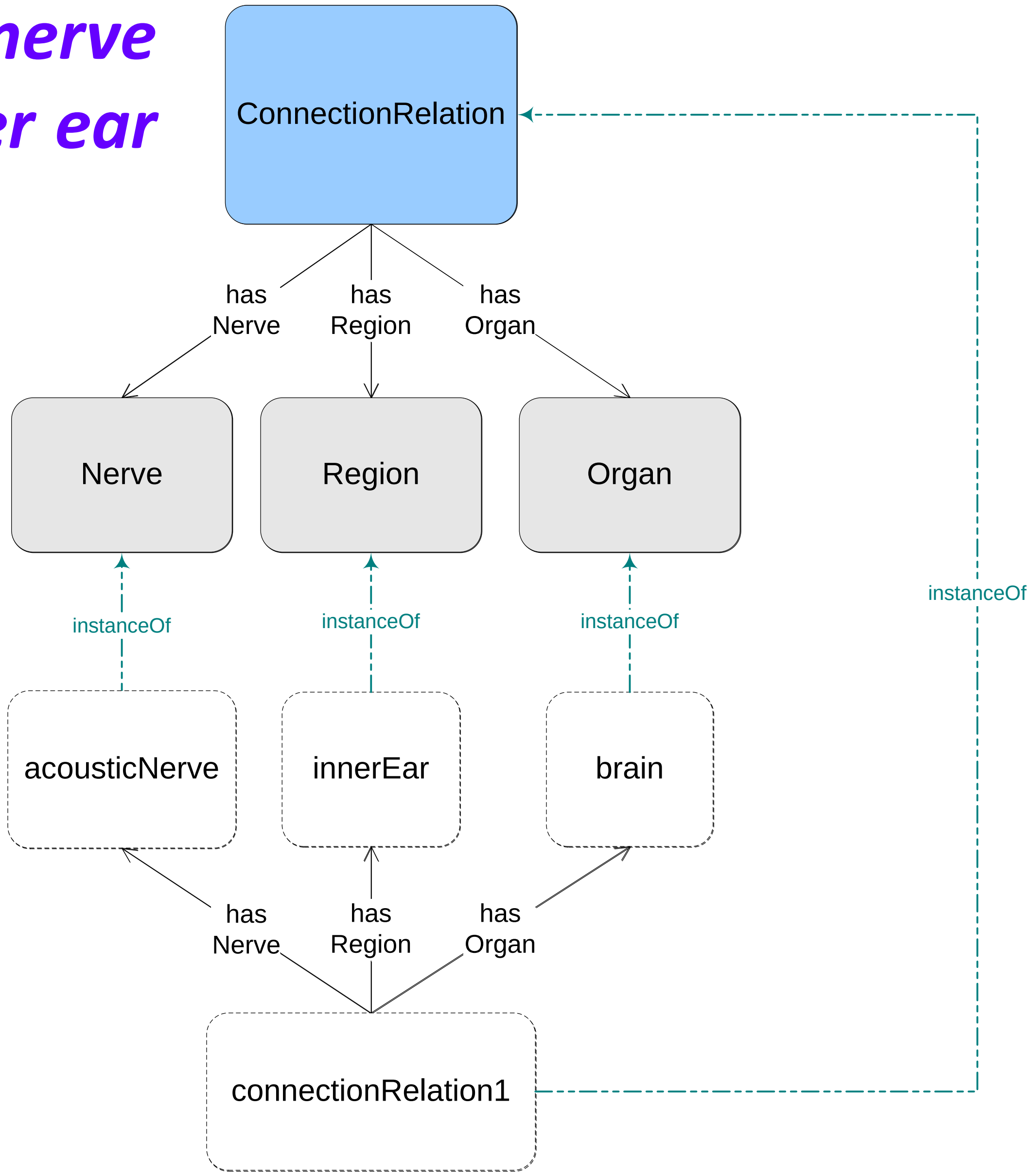
Pattern 3: Several Individuals in a Relation. Pattern 3b: Connection Relation

- **(General) Situation: A nerve connect a region of the body with an organ of the nervous system**
  - **(Specific) Situation**: *The acoustic nerve connects the inner ear and the brain*
    - **Functional Representation: *ConnectionRelation (Nerve, Region, Organ)***

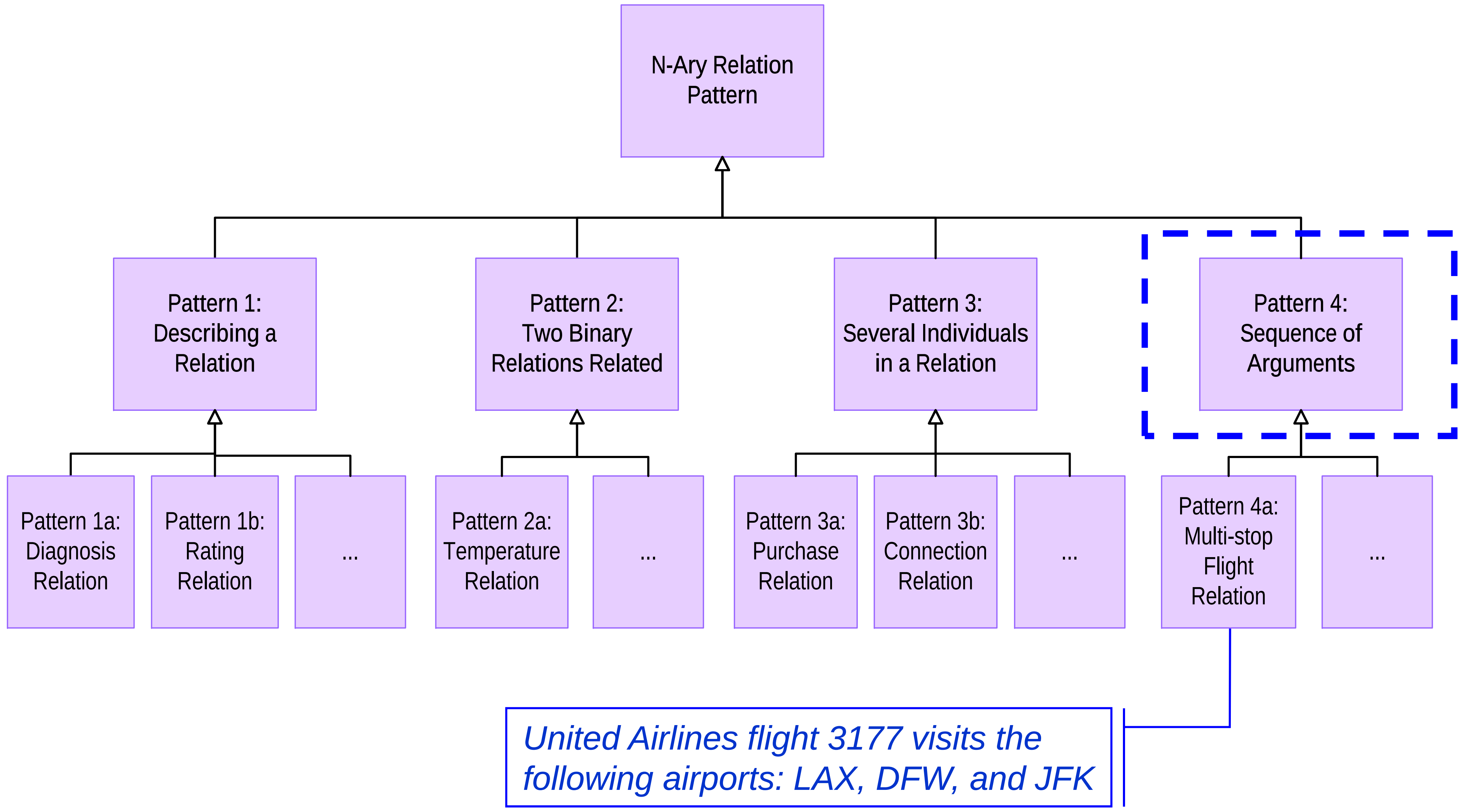


Pattern 3: Several Individuals in a Relation. Pattern 3b: Connection Relation

- ***The acoustic nerve connects the inner ear and the brain***



Specializations of the N-ary Relation Pattern

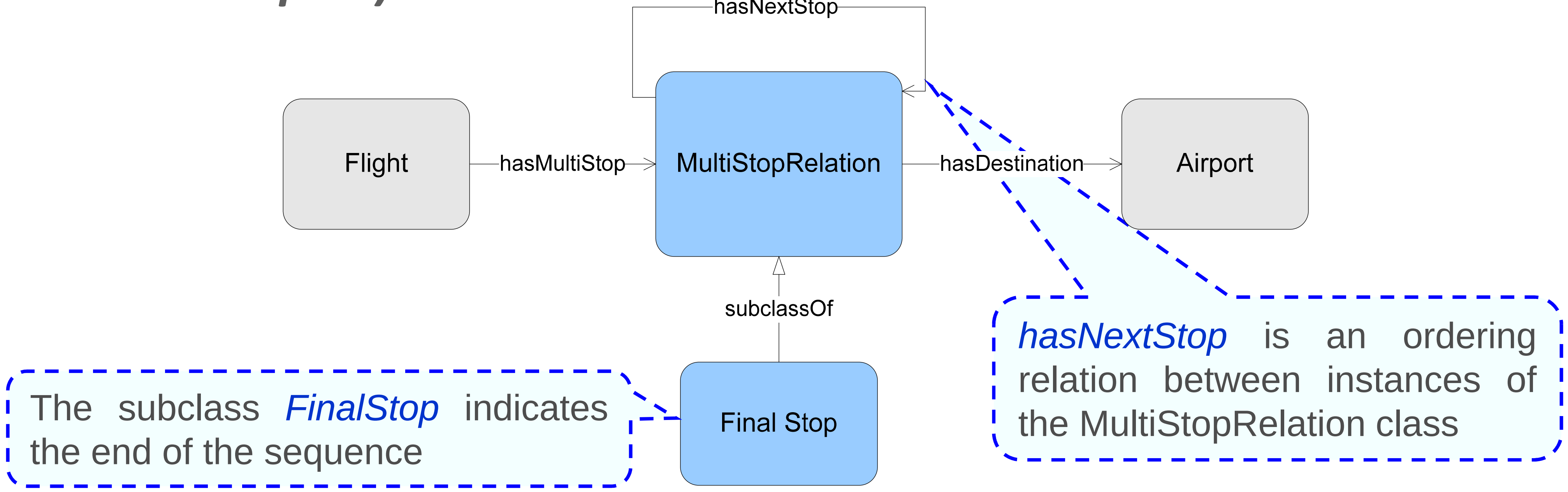

N-Ary Relation Pattern
Pattern 1: Describing a Relation
Pattern 2: Two Binary Relations Related
Pattern 3: Several Individuals in a Relation
Pattern 4: Sequence of Arguments
Pattern 1a: Diagnosis Relation
Pattern 1b: Rating Relation
...
Pattern 2a: Temperature Relation
...
Pattern 3a: Purchase Relation
Pattern 3b: Connection Relation
...
Pattern 4a: Multi-stop Flight Relation
...
United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK

Pattern 4: Sequence of Arguments

- **Linking from, or to, an ordered list of individuals**: using lists for arguments in a relation
  - *United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK*
- Some n-ary relations are similar to a list or sequence of arguments
  - These relations might hold between many different numbers of arguments, and there is no natural way to break it up into a set of distinct properties. The order of the arguments is highly meaningful
- **Basic idea**: when all but one participant in a relation do not have a specific role and essentially form an ordered list, it is natural to connect these arguments into a sequence according to some relation, and to relate the one participant to this sequence (or the first element of the sequence)

Pattern 4: Sequence of Arguments. Pattern 4a: Multi-stop Flight Relation

- **(General) Situation: A particular flight has multiple stops in different airports**
  - **(Specific) Situation**: *United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK*
    - **Functional Representation:** ***MultiStopRelation (Flight, Airport)***

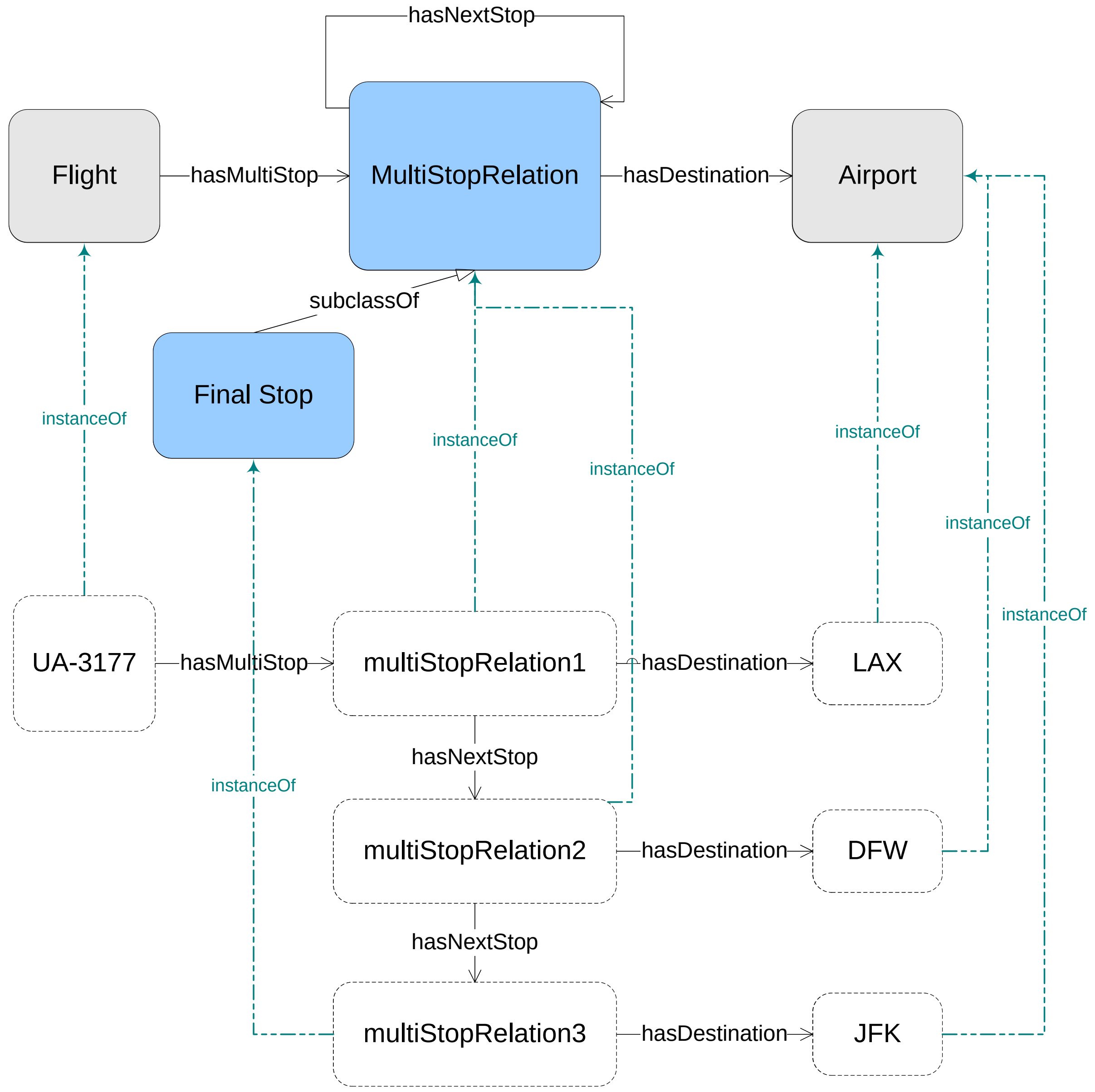


hasNextStop is an ordering relation between instances of the MultiStopRelation class

The subclass *FinalStop* indicates the end of the sequence

Pattern 4: Sequence of Arguments. Pattern 4a: Multi-stop Flight Relation

- **United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK**



Index

Ontology Design Patterns

Ontology Design Pattern (ODP) is a modeling solution to solve a recurrent ontology design problem

- Pattern is associable with the wider "**good/best practice**" of software engineering. It includes a wider range of solution types. For example: naming conventions in software engineering are considered good practices, they are not design patterns
- ODPs can be classified into **six families**
  - Each family addresses different kinds of problems, and can be represented with different levels of formality

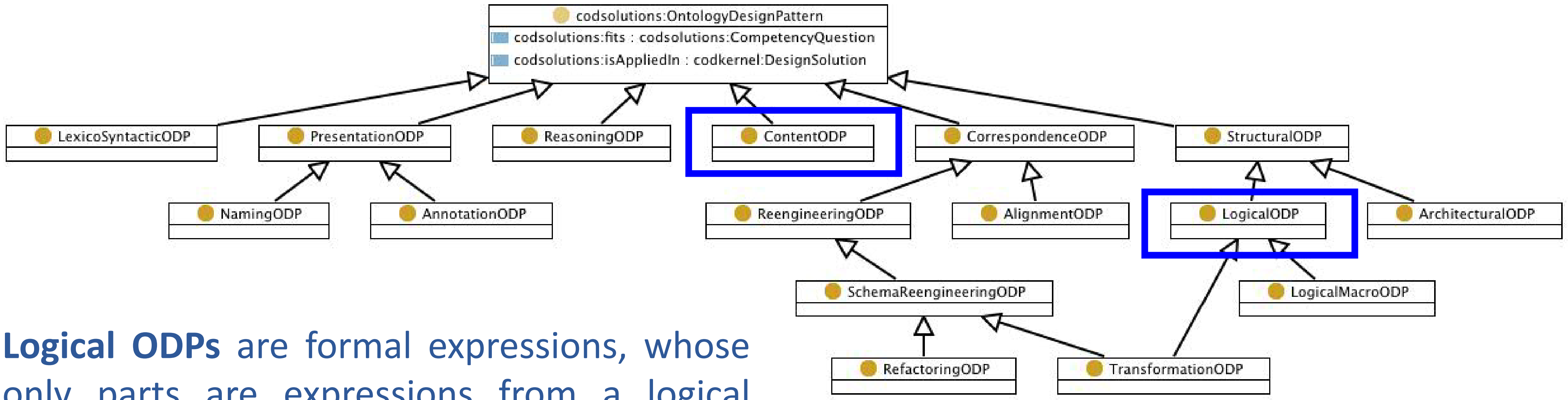
The **goal of the ODPs reuse** is

- to facilitate the solution of modelling issues
- to improve interoperability through using well-proven solutions and best practices, in the form of patterns

The **idea of applying patterns for modelling ontologies** was proposed by [Clark et al., 2000]

Clark, P., Thompson, J., & Porter, B. W. *Knowledge Patterns*. In KR2000: Principles of Knowledge Representation and Reasoning. pp. 591-600. 2000

Types of Ontology Design Patterns



Logical ODPs are formal expressions, whose only parts are expressions from a logical vocabulary e.g., OWL DL, that solve a problem of expressivity

- Logical ODPs are independent from a specific domain of interest, i.e. they are content-independent
- Logical ODPs solve design problems where the primitives of the representation language do not directly support certain logical constructs

(E.g., N-Ary Relation)

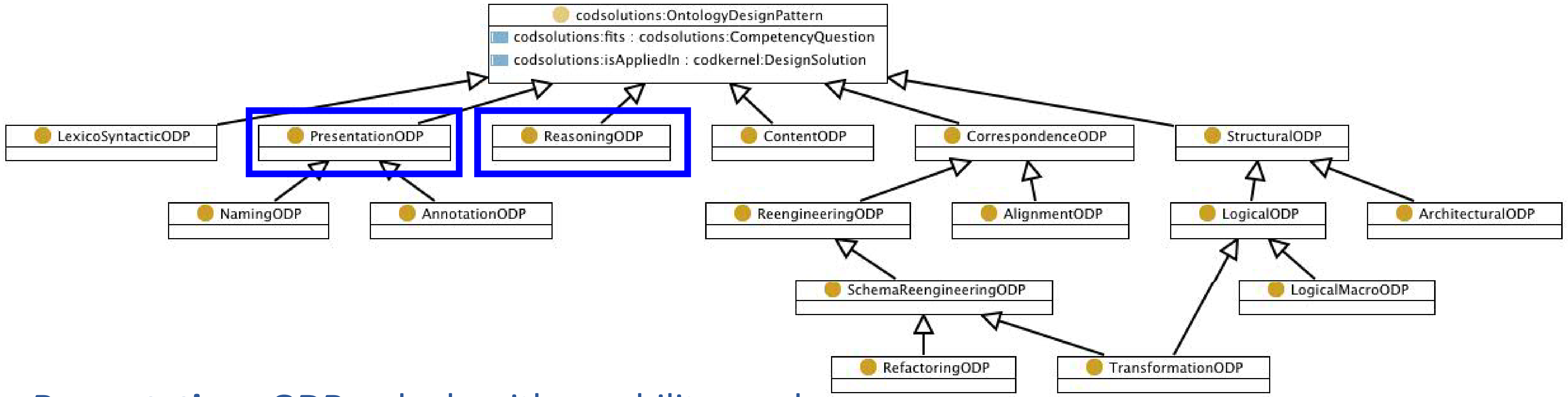
Content OPs (CPs) encode conceptual, rather than logical design patterns

- Logical OPs solve design problems independently of a particular conceptualization
- CPs propose patterns for solving design problems for the domain classes and properties that populate an ontology. They address content problems

(E.g. Agent-Role)

http://ontologydesignpatterns.org/wiki/OPTypes

Types of Ontology Design Patterns

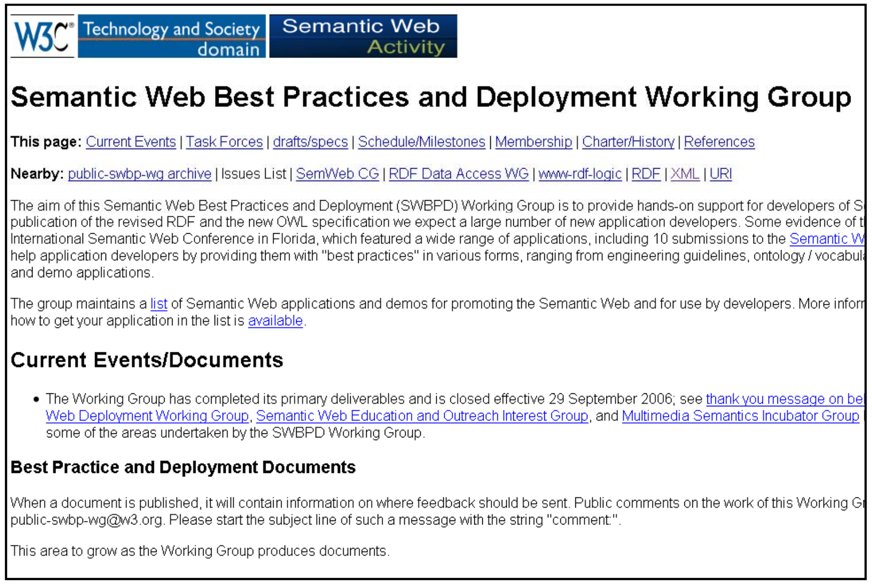


Presentation ODPs deal with usability and readability of ontologies from a user perspective

Reasoning ODPs are applications of Logical ODPs oriented to obtain certain reasoning results, based on the behavior implemented in a reasoning engine

http://ontologydesignpatterns.org/wiki/OPTypes

Ontology Design Patterns: Catalogues



W3C Technology and Society domain | Semantic Web Activity

Semantic Web Best Practices and Deployment Working Group

This page: Current Events | Task Forces | drafts/specs | Schedule/Milestones | Membership | Charter/History | References

Nearby: public-swbp-wg archive | Issues List | SemWeb CG | RDF Data Access WG | www-rdf-logic | RDF | XML | URI

The aim of this Semantic Web Best Practices and Deployment (SWBPD) Working Group is to provide hands-on support for developers of S
publication of the revised RDF and the new OWL specification we expect a large number of new application developers. Some evidence of t
International Semantic Web Conference in Florida, which featured a wide range of applications, including 10 submissions to the Semantic W
help application developers by providing them with "best practices" in various forms, ranging from engineering guidelines, ontology / vocabul
and demo applications.

The group maintains a list of Semantic Web applications and demos for promoting the Semantic Web and for use by developers. More infor
how to get your application in the list is available.

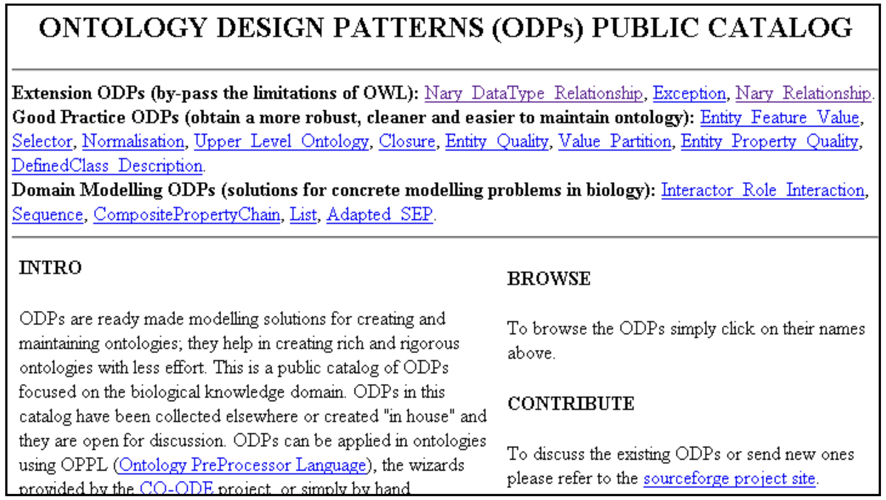
Current Events/Documents

- The Working Group has completed its primary deliverables and is closed effective 29 September 2006; see thank you message on be
Web Deployment Working Group, Semantic Web Education and Outreach Interest Group, and Multimedia Semantics Incubator Group
some of the areas undertaken by the SWBPD Working Group.

Best Practice and Deployment Documents

When a document is published, it will contain information on where feedback should be sent. Public comments on the work of this Working G
public-swbp-wg@w3.org. Please start the subject line of such a message with the string "comment:".

This area to grow as the Working Group produces documents.



ONTOLOGY DESIGN PATTERNS (ODPs) PUBLIC CATALOG

Extension ODPs (by-pass the limitations of OWL): Nary_DataType_Relationship, Exception, Nary_Relationship.
Good Practice ODPs (obtain a more robust, cleaner and easier to maintain ontology): Entity_Feature_Value, Selector, Normalisation, Upper_Level_Ontology, Closure, Entity_Quality, Value_Partition, Entity_Property_Quality, DefinedClass_Description.
Domain Modelling ODPs (solutions for concrete modelling problems in biology): Interactor_Role_Interaction, Sequence, CompositePropertyChain, List, Adapted_SEP.

INTRO

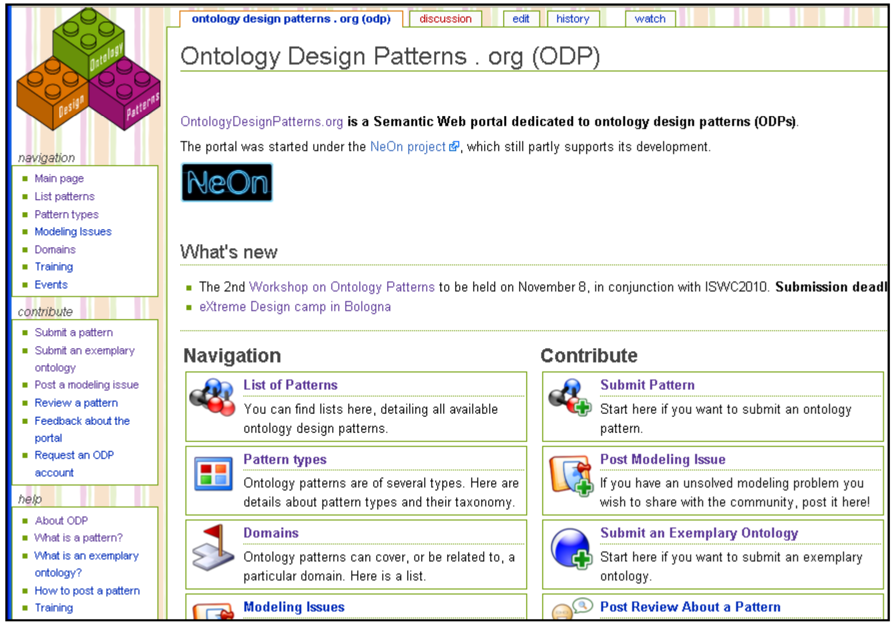
ODPs are ready made modelling solutions for creating and maintaining ontologies; they help in creating rich and rigorous ontologies with less effort. This is a public catalog of ODPs focused on the biological knowledge domain. ODPs in this catalog have been collected elsewhere or created "in house" and they are open for discussion. ODPs can be applied in ontologies using OPPL (Ontology PreProcessor Language), the wizards provided by the CO-ODE project, or simply by hand.

BROWSE

To browse the ODPs simply click on their names above.

CONTRIBUTE

To discuss the existing ODPs or send new ones please refer to the sourceforge project site.



ontology design patterns . org (odp) | discussion | edit | history | watch

Ontology Design Patterns . org (ODP)

OntologyDesignPatterns.org **is a Semantic Web portal dedicated to ontology design patterns (ODPs)**.

The portal was started under the NeOn project, which still partly supports its development.

NeOn

What's new

- The 2nd Workshop on Ontology Patterns to be held on November 8, in conjunction with ISWC2010. **Submission deadl**
- eXtreme Design camp in Bologna

Navigation

List of Patterns
You can find lists here, detailing all available ontology design patterns.

Pattern types
Ontology patterns are of several types. Here are details about pattern types and their taxonomy.

Domains
Ontology patterns can cover, or be related to, a particular domain. Here is a list.

Modeling Issues

Contribute

Submit Pattern
Start here if you want to submit an ontology pattern.

Post Modeling Issue
If you have an unsolved modeling problem you wish to share with the community, post it here!

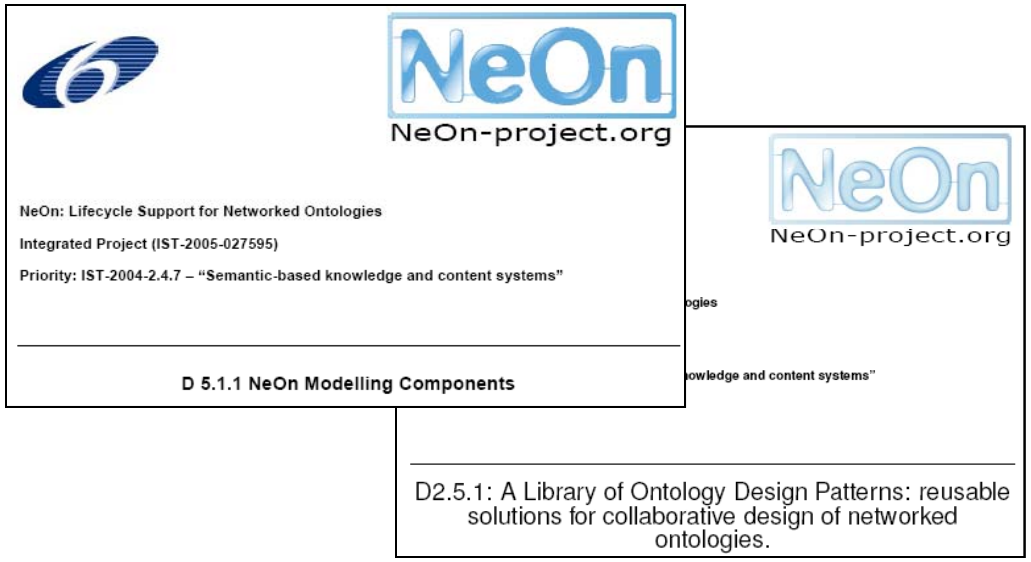
Submit an Exemplary Ontology
Start here if you want to submit an exemplary ontology.

Post Review About a Pattern

navigation
- Main page
- List patterns
- Pattern types
- Modeling Issues
- Domains
- Training
- Events

contribute
- Submit a pattern
- Submit an exemplary ontology
- Post a modeling issue
- Review a pattern
- Feedback about the portal
- Request an ODP account

help
- About ODP
- What is a pattern?
- What is an exemplary ontology?
- How to post a pattern
- Training



NeOn
NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D 5.1.1 NeOn Modelling Components

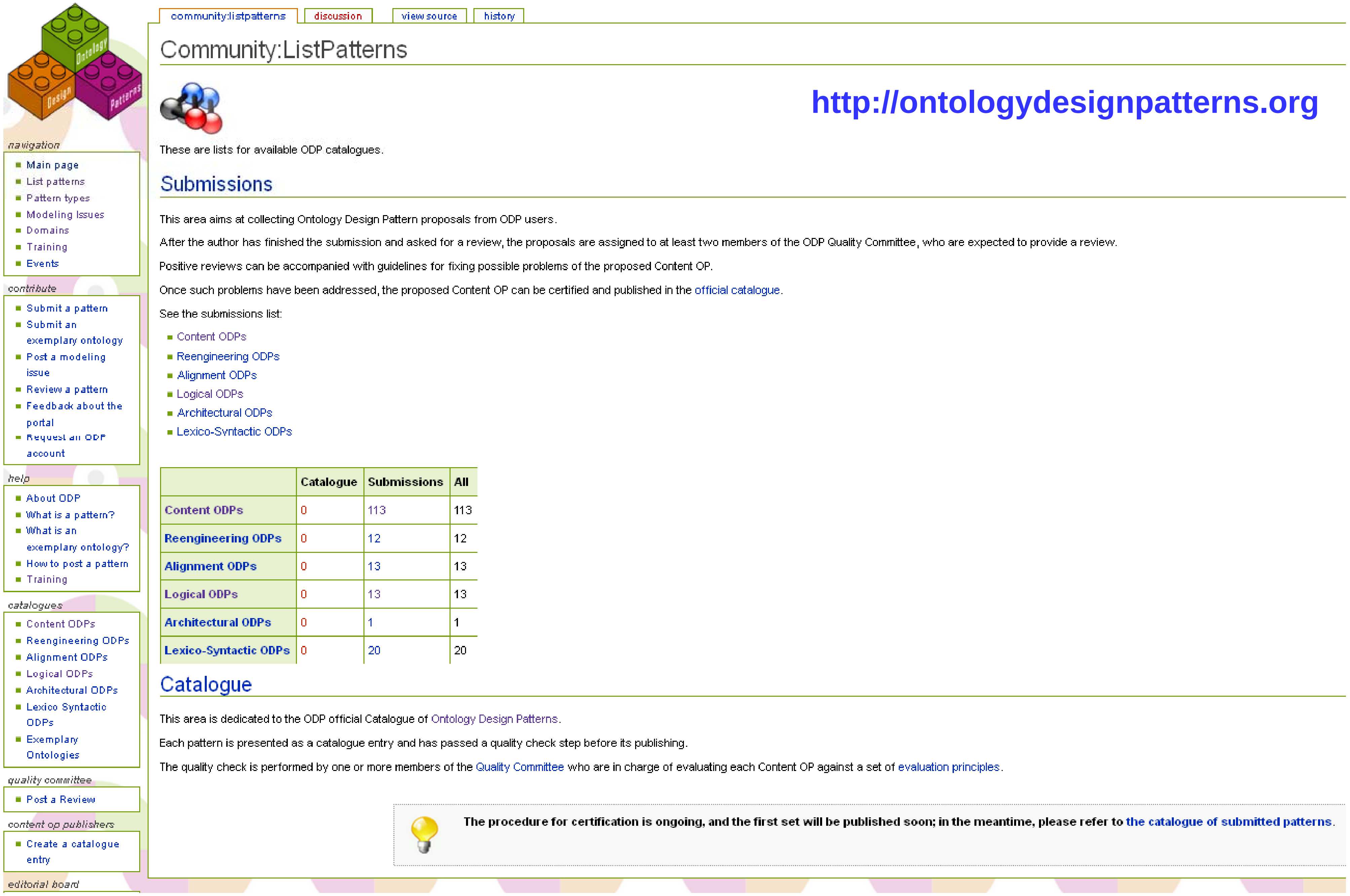
NeOn
NeOn-project.org

ogies

owledge and content systems"

D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies.

Inventory of ODPs



community:listpatterns | discussion | view source | history

navigation
- Main page
- List patterns
- Pattern types
- Modeling Issues
- Domains
- Training
- Events

contribute
- Submit a pattern
- Submit an exemplary ontology
- Post a modeling issue
- Review a pattern
- Feedback about the portal
- Request an ODP account

help
- About ODP
- What is a pattern?
- What is an exemplary ontology?
- How to post a pattern
- Training

catalogues
- Content ODPs
- Reengineering ODPs
- Alignment ODPs
- Logical ODPs
- Architectural ODPs
- Lexico Syntactic ODPs
- Exemplary Ontologies

quality committee
- Post a Review

content op publishers
- Create a catalogue entry

editorial board

Community:ListPatterns

http://ontologydesignpatterns.org

These are lists for available ODP catalogues.

Submissions

This area aims at collecting Ontology Design Pattern proposals from ODP users.

After the author has finished the submission and asked for a review, the proposals are assigned to at least two members of the ODP Quality Committee, who are expected to provide a review.

Positive reviews can be accompanied with guidelines for fixing possible problems of the proposed Content OP.

Once such problems have been addressed, the proposed Content OP can be certified and published in the official catalogue.

See the submissions list:

- Content ODPs
- Reengineering ODPs
- Alignment ODPs
- Logical ODPs
- Architectural ODPs
- Lexico-Syntactic ODPs

	Catalogue	Submissions	All
Content ODPs	0	113	113
Reengineering ODPs	0	12	12
Alignment ODPs	0	13	13
Logical ODPs	0	13	13
Architectural ODPs	0	1	1
Lexico-Syntactic ODPs	0	20	20

Catalogue

This area is dedicated to the ODP official Catalogue of Ontology Design Patterns.

Each pattern is presented as a catalogue entry and has passed a quality check step before its publishing.

The quality check is performed by one or more members of the Quality Committee who are in charge of evaluating each Content OP against a set of evaluation principles.

The procedure for certification is ongoing, and the first set will be published soon; in the meantime, please refer to the catalogue of submitted patterns.

Use Case: Chess Domain

Competency Questions

1. Who played against Kasparov in the round 1994 Lineares tournament? Did (s)he play as a white or black player?
2. What is the first move taken by the black player in the Sicilian Defense opening?
3. Find all games in which Bobby Fischer, playing black, lost in the poisoned pawn variation of the Sicilian Defence opening.
4. Are there any recorded games using the Grünfeld Defence from before the 20th century?
5. What did Kasparov say about his opponent’s first two moves in his commentary about his game against Topalov in the 1999 Tournament in Wijk ann Zee?
6. Who was the first non-Russian world champion after Fischer?
7. Did Bobby Fischer ever play against a grandmaster in Germany?
8. List all world championship games won by forfeit.

Use Case: Chess Domain

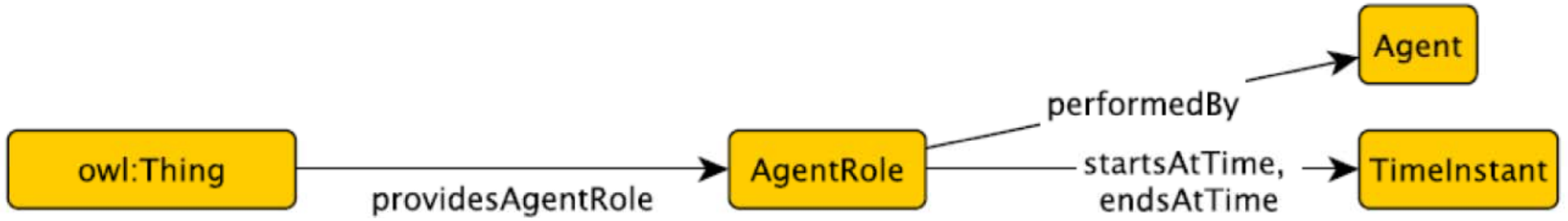
Analysis

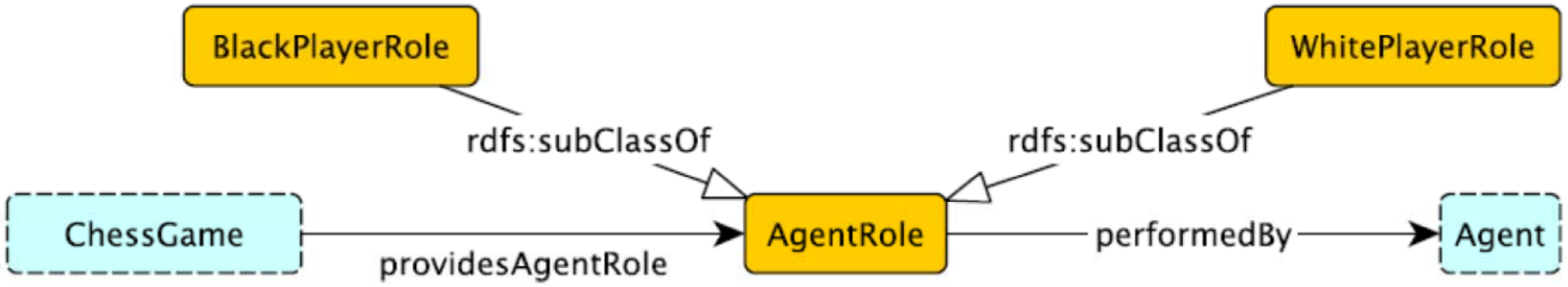
- **Understand the nature of the things you are modeling.**

Chess game	**...**	**An Event**
Half-move	**…**	**A Subevent of a chess game**
Player	**…**	**The Role of an Agent**
Opening	**…**	**this is probably complex**
tournaments	**…**	**Events**
commentary	**…**	**this is again more complex**

Use Case: Chess Domain

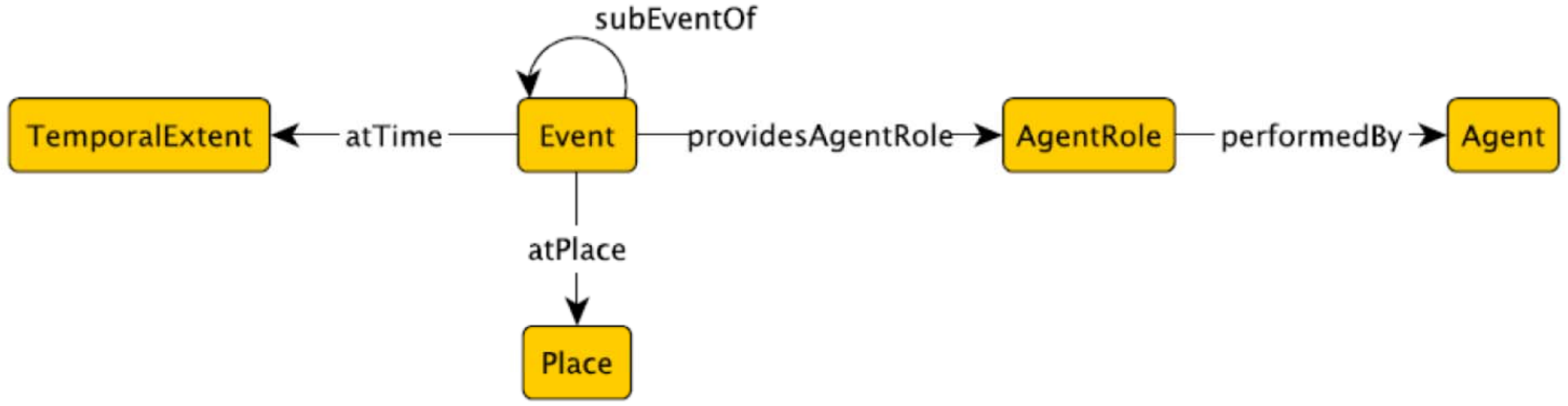
Player as AgentRole

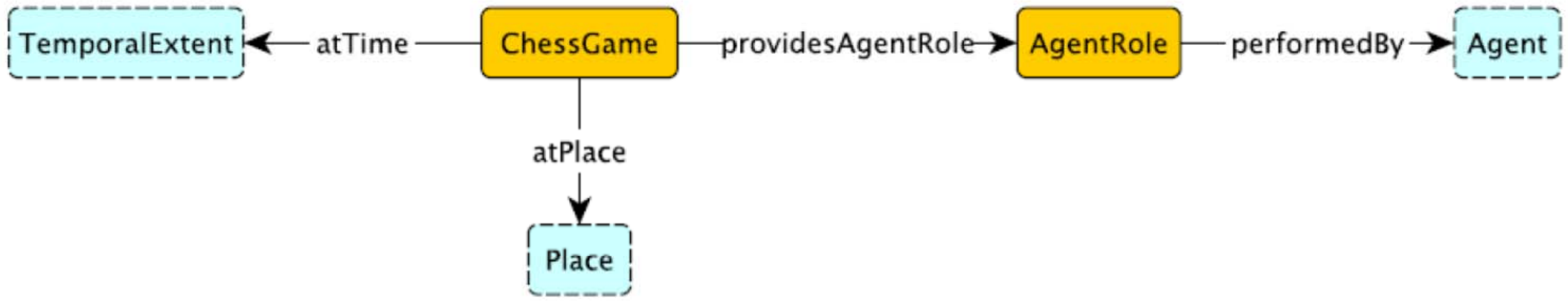




Use Case: Chess Domain

ChessGame as Event





Mini-Exercise 1

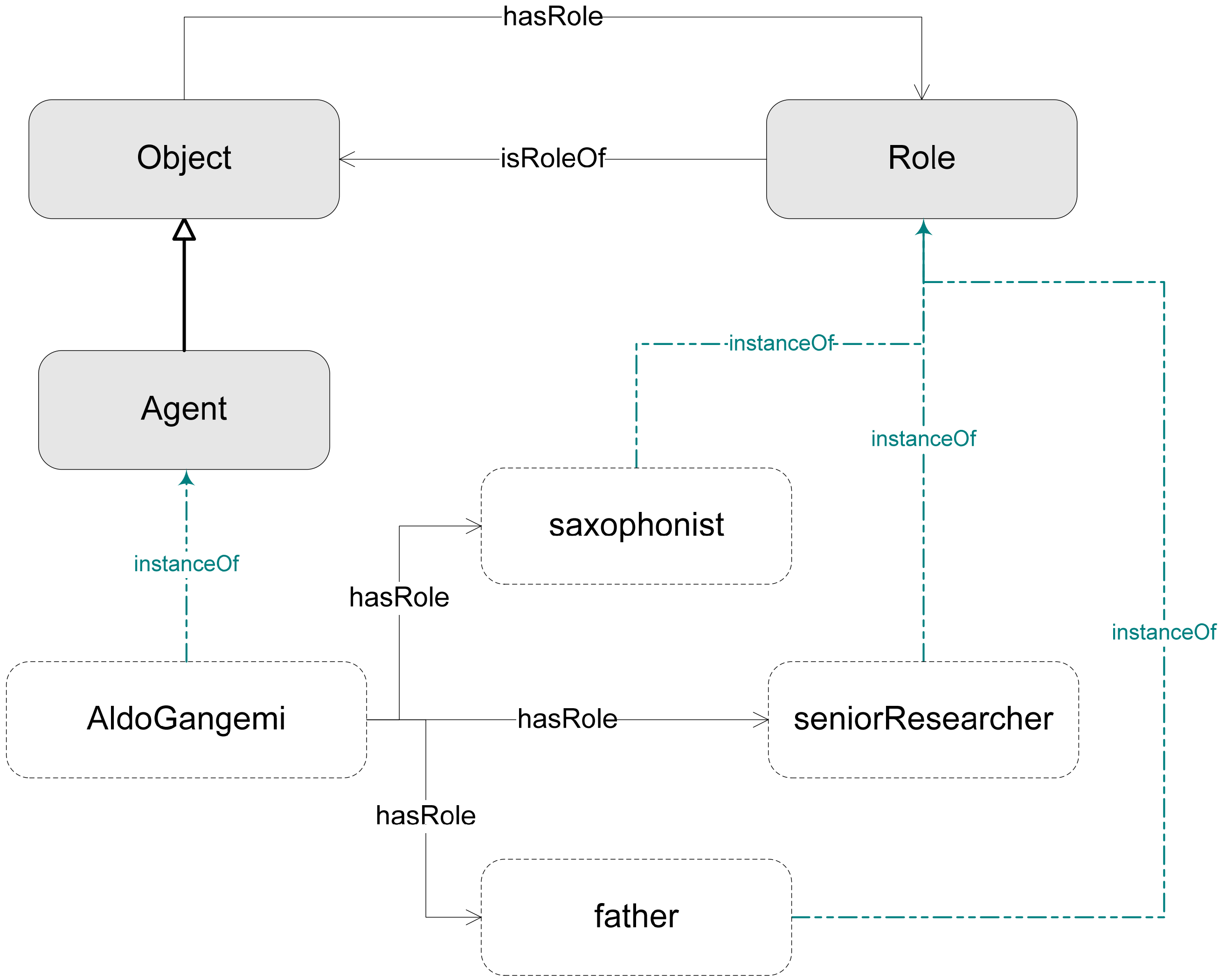
We are going to **model** the following problema/scenario:

- Aldo Gangemi is a senior researcher. He is also father and a saxophonist

Mini-Exercise 1

We are going to **model** the following problema/scenario:

- Aldo Gangemi is a senior researcher. He is also father and a saxophonist



Mini-Exercise 2

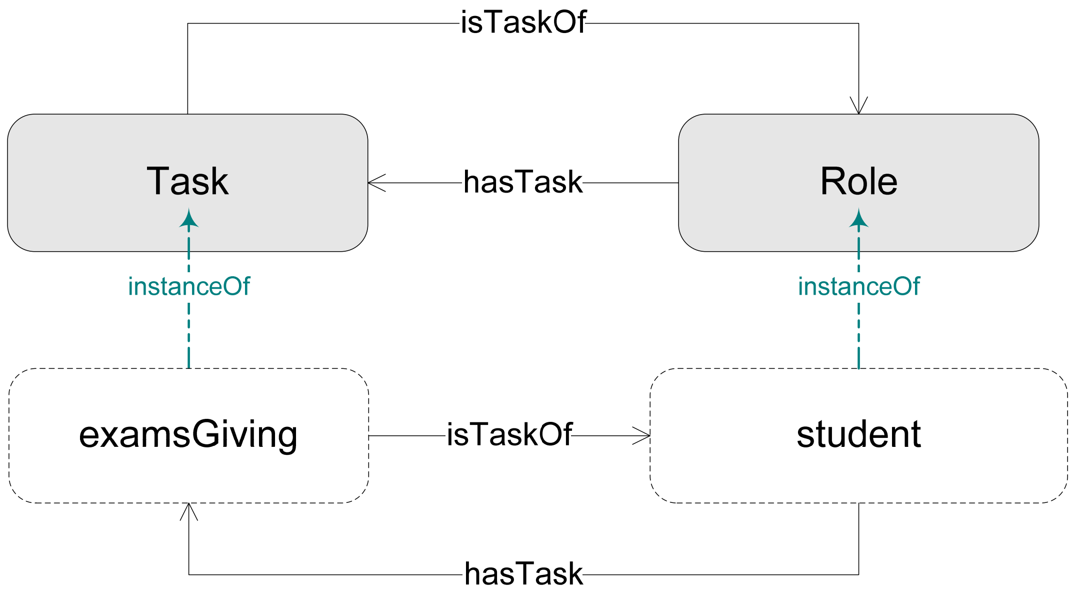
We are going to **model** the following problema/scenario:

- Students have the duty of giving exams

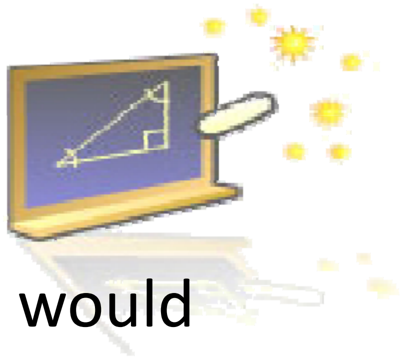
Mini-Exercise 2

We are going to **model** the following problema/scenario:

- Students have the duty of giving exams



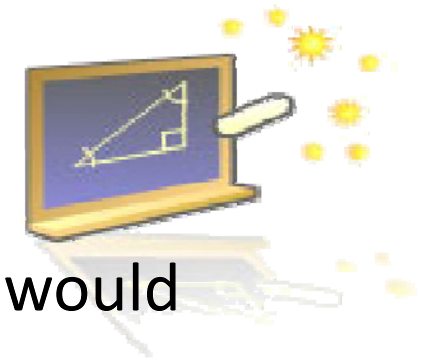
Mini-Exercise 3



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following sentence:

- The fifth track of “Thriller” is “Beat it”.

Mini-Exercise 3



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following sentence:

- The fifth track of “Thriller” is “Beat it”.

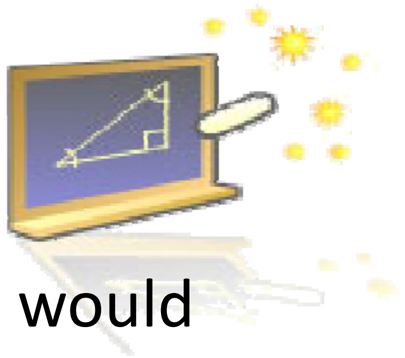
Patterns that can be reused:

- Information Realization
- Part of

Explanation:

- "Beat it" is an information object, while a track is one of different possible realizations of it.
    - Note: It is convenient to specialize them with domain-specific terminology e.g. Track rdfs:subClassOf InformationRealization.
- "Thriller" is an information object. This sentence refers to a specific recording of it, hence its realization.
- The album is composed of a number of tracks, hence you need to represent this relation by part of.

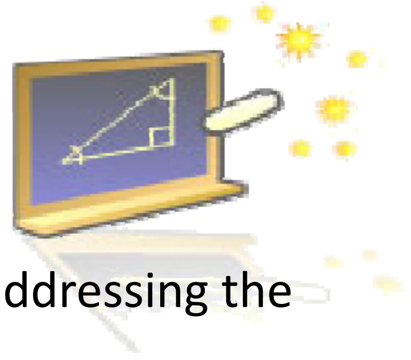
Mini-Exercise 4



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following sentence:

- Aldo Gangemi and Jerome Euzenat were general chairs of EKAW 2008, which they attended for the whole duration.

Mini-Exercise 4



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following sentence:

- Aldo Gangemi and Jerome Euzenat were general chairs of EKAW 2008, which they attended for the whole duration.

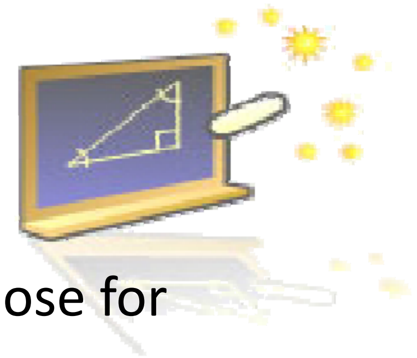
Patterns that can be reused:

- Participation
- Object Role
- Situation

Explanation:

- In this sentence there is a clear reference to the role played by Aldo and Jerome, hence the relationship between a person and its role is needed. Notice that the relation of playing a role is mentioned with respect to a specific event i.e. EKAW 2008.
  - This means that the sentence expresses an n-ary relation between a person, the role it plays, and in which event. In order to capture this we can compose situation with object role.
  - Note: It is convenient to specialize the selected patterns with domain-specific terminology e.g. ConferenceRole rdfs:subClassOf Role.
- Furthermore, the sentence mentions their attendance to an event i.e., EKAW 2008, which means they participate in it. You can use the participation pattern.

Mini-Exercise 5



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following competency questions:

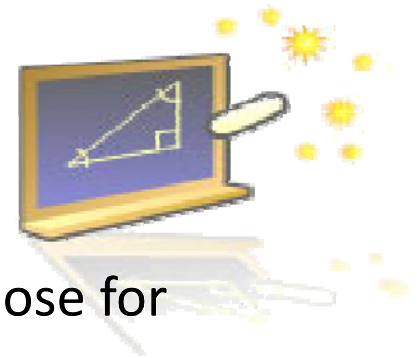
1. Which burglaries were performed from 2016-03-01 through 2016-03-10 in the town of Ipswitch?
2. Which persons were involved in the above burglaries in any role (as victims, suspects, or known offenders)?
3. Which persons are known to have been responsible for any burglaries?
4. Which of the above known burglars were observed in the town of Ipswitch during February or May of 2016?
5. Which dates of birth have been attributed to the person who calls himself ”Johnny Burglar” and lives on ”Burglarroad 1”?
6. Which other addresses have been attributed to the person who goes by that name?
7. Which persons are known to have committed burglaries jointly with Johnny Burglar?
8. Bernard Madoff is the known offender in a case of fraud. Which other people were involved in that same crime?
9. In 2010 Johnny Burglar was on vacation in Torquay. Did any burglaries take place in Torquay during the time he stayed there?

Mini-Exercise 5

Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following competency questions:

- **Recommended ODPs**:
  - Place
  - ParticipantRole
  - Time Interval

Mini-Exercise 6: Theater Productions



Which Content Pattern (CP) (or combination of CPs) you would choose for addressing the following competency questions:

1. When did a certain theatre festival take place?
2. Where did a certain festival take place?
3. What plays could be seen during a certain theatre festival?
4. In what city is a certain theatre located?
5. In what country is a certain city located?
6. What play is the basis of this production?
7. During what time period was a certain play written?
8. What is the “star rating” given by a certain newspaper for a certain production?
9. At what time did a certain actor start working for a specific theatre?
10. What roles does a certain person have within a certain production at a certain point in time?

Mini-Exercise 6: Theater Productions

There can be alternative solutions, however, one possible set of ODPs could be:

- **CQs 1-3**: **Situation** - The festival is a specialization of a Situation, and in its setting there is a place, a time, and a number of plays (connected by specializing the hasSetting/isSettingFor properties + restrictions on the festival class).
- **CQs 4-5**: **Place** - Both cities and countries are places, and the hasLocation/isLocationOf properties can be used for both these cases + restrictions on the classes, e.g. so that cities cannot be located in other cities but only in countries etc. In addition the pattern specialization needs to be extended with a theater class.
- **CQ 6**: **Information Realization** - The production can be seen as a concrete realization of the abstract play, and each production is the realization of exactly one play.
- **CQ 7**: **Time Interval** - Creating an instance of a time interval class, rather than just two literal values allows us to talk about the interval as such, and set restrictions on it, e.g. that each play is written during exactly one interval, and that interval in turn has exactly one start and one end point.
- **CQ 8, 9, 10**: **Situation** - Again, these CQs can each be seen as a "situation", i.e. an n-ary relation, with several things involved in the setting. In fact, we need them to be modeled in this way in order to be able to distinguish the particular combination of parameters.

Acknowledgments

- Asunción Gómez-Pérez (OEG)
- María Poveda-Villalón (OEG)
- Valentina Presutti (CNR)
- Valentina Tamma (University of Liverpool)

- Some of the exercises have been taken from: "http://ontologydesignpatterns.org/wiki/Training:Main"

Course: Intelligent Systems

Unit 3: Ontology Engineering

Patterns in Knowledge Representation

Mari Carmen Suárez de Figueroa Baonza

Course 2022 – 2023

Technical University of Madrid

Ontology Requirements: Definition

- **Ontology requirements** are those needs that the ontology to be built should represent
- **Competency Questions (CQs)** are questions that the ontology to be built should be able to answer
  - CQs are a way to **represent** ontology requirements
  - CQs can be written in **natural language** (NL)
  - CQs can be formalized in **ontology query languages** (e.g., SPARQL)

Ontology Requirements: Types

- **Ontology requirements** can be divided into two types
  - **Non-functional requirements**
    - They refer to general aspects not related to the content that the ontology should represent
  - **Functional requirements**
    - They refer to the particular knowledge that the ontology should represent
    - They are also known as content requirements
    - They should be written as competency questions with their corresponding response

Ontology Requirements: Functional Requirements

- Functional requirements should be written as **competency questions with their corresponding response**
- Example:
  - *Question*: Where can be located a temperature sensor?
  - *Response*: A temperature sensor can be located either in a mobile or in street furniture

Ontology Requirements: Functional Requirements

- In some cases, functional requirements can be written as **sentences in natural languages**
  - These sentences are also called general characteristics
    - Example: A microphone is characterized by frequency response, output impedance, sensitivity (at 1 kHz, open circuit voltage), weight, and type
  - These sentences will be later transformed to competency questions with their response
    - Example:
      - *Question*: What are the characteristics of a microphone?
      - *Response*: Frequency response, output impedance, sensitivity (at 1 kHz, open circuit voltage), weight, and type