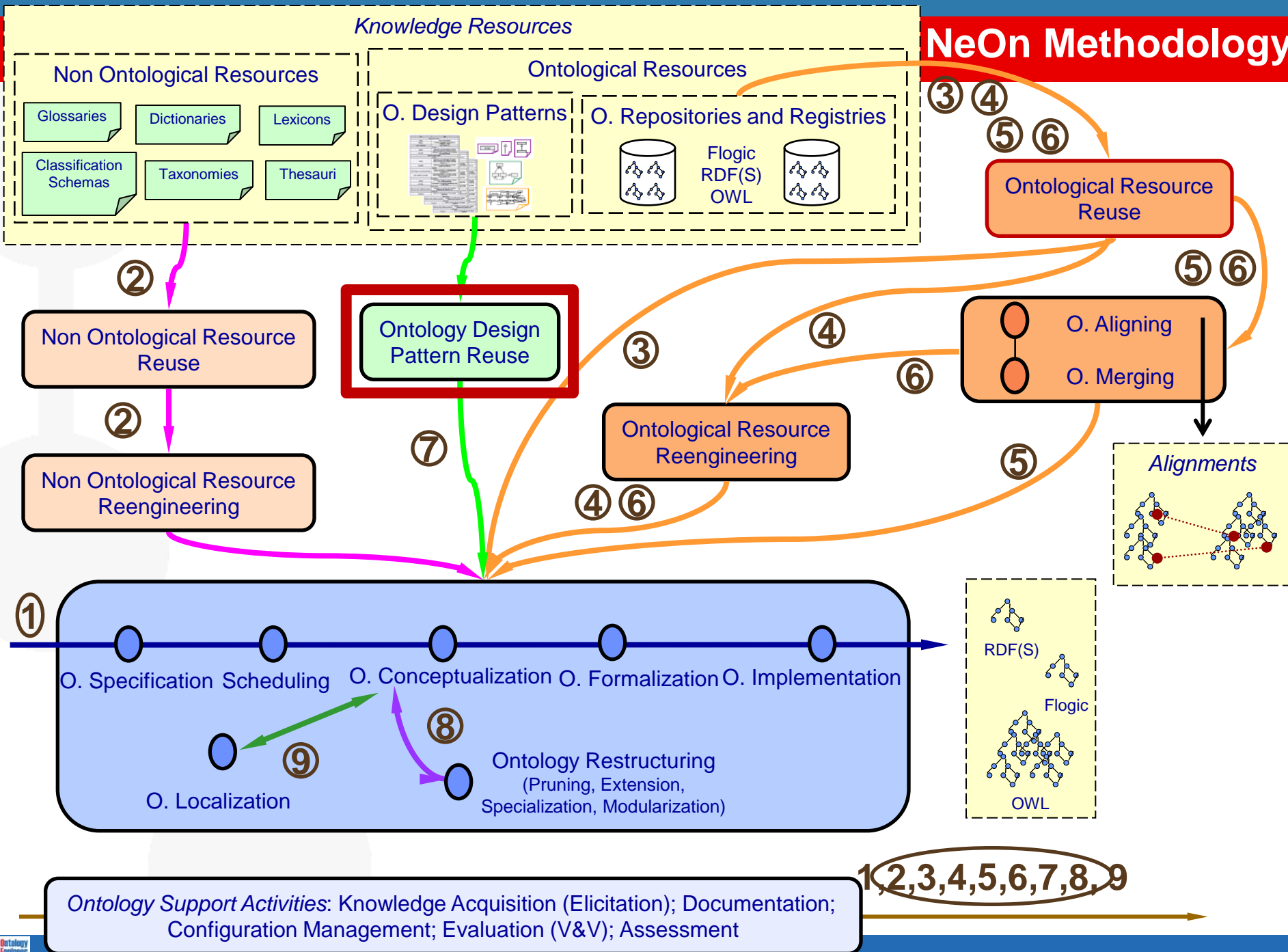




Methodological Guidelines for Reusing Ontology Design Patterns

Mari Carmen Suárez-Figueroa

NeOn Methodology




- ❑ **Ontology Design Patterns (ODPs): Introduction**
- ❑ Methodological Guidelines for Reusing ODPs

- **Pattern** is something proposed for imitation.
- **Design Pattern** refers to shared guidelines that help solve design problems.
- **Ontology Design Pattern (ODP)** is a modeling solution to solve a recurrent ontology design problem.
 - How to represent the following problems?
 - *“Mariano Fernández-López and Mari Carmen Suárez-Figueroa are senior researchers. Mariano is also associate professor and Mari Carmen is teaching assistant”*
 - *“Mariano and Mari Carmen co-participate at the CAEPIA 2007 conference”*
 - 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

Ontology Design Patterns: Catalogues (I)



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.rdflogic](#) | [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 the International Semantic Web Conference in Florida, which featured a wide range of applications, including 10 submissions to the [Semantic Web](#) help application developers by providing them with "best practices" in various forms, ranging from engineering guidelines, ontology / vocabulary 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 information 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 behalf of the 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 Group should be sent to 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	BROWSE
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.	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)


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.



What's new

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

Navigation	Contribute
 List of Patterns You can find lists here, detailing all available ontology design patterns.	 Submit Pattern Start here if you want to submit an ontology pattern.
 Pattern types Ontology patterns are of several types. Here are details about pattern types and their taxonomy.	 Post Modeling Issue If you have an unsolved modeling problem you wish to share with the community, post it here!
 Domains Ontology patterns can cover, or be related to, a particular domain. Here is a list.	 Submit an Exemplary Ontology Start here if you want to submit an exemplary ontology.
 Modeling Issues	 Post Review About a Pattern




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

NeOn: Lifecycle Support for Networked Ontologies
Integrated Project (IST-2005-027595)
Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

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



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

[community:listpatterns](#)

[discussion](#)

[view source](#)

[history](#)

Community:ListPatterns



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, review.

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

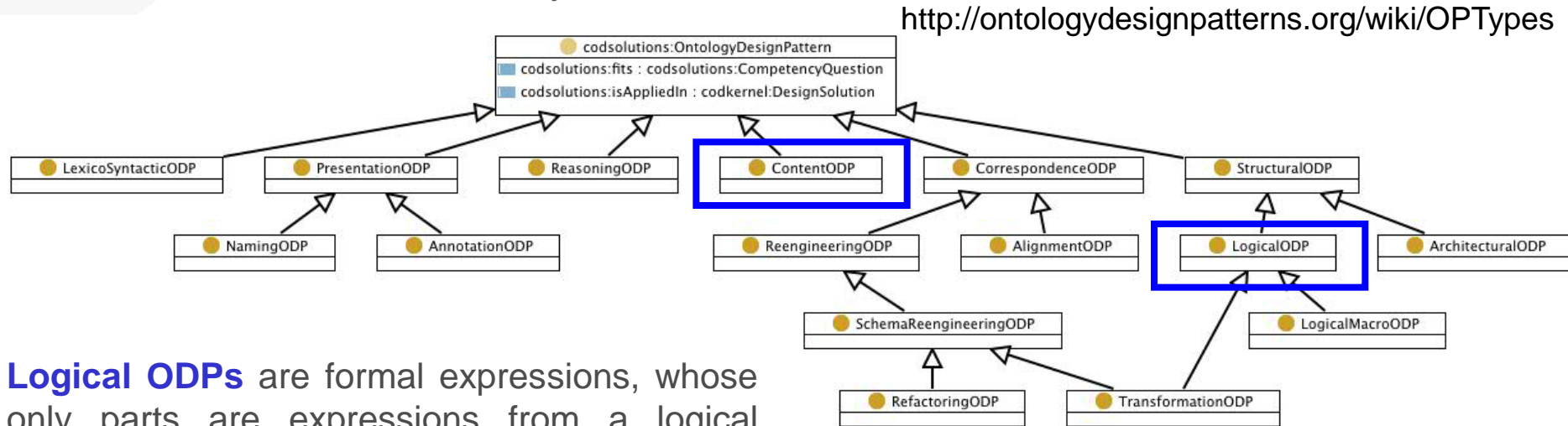
Once such problems have been addressed, the proposed Content OP can be certified and published in the [official catalogue](#).

See the submissions list:

- [Content ODPs](#)
- [Reengineering ODPs](#)
- [Alignment ODPs](#)
- [Logical ODPs](#)
- [Architectural ODPs](#)
- [Lexico-Syntactic ODPs](#)

Types of Ontology Design Patterns (I)

- ODPs can be classified into **six families**
 - Each family addresses different kinds of problems, and can be represented with different levels of formality.



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

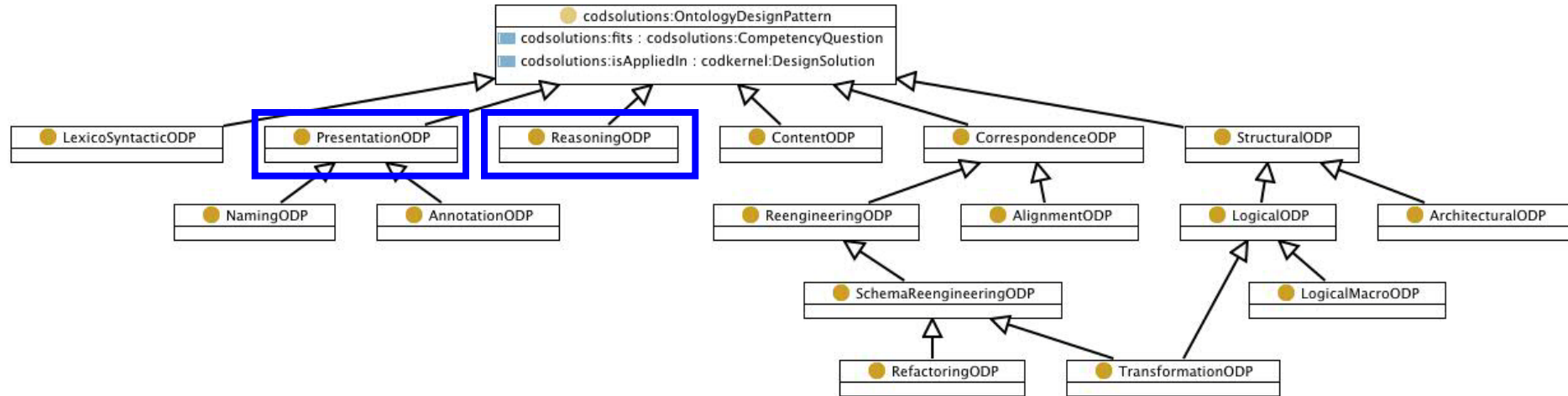
(E.g., *N-Ary Relation*)

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

- 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*)

Types of Ontology Design Patterns (II)



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

- Ontology Design Patterns (ODPs): Introduction
- **Methodological Guidelines for Reusing ODPs**

ODPs Reuse: Methodological Guidelines (I)

Ontology Design Pattern Reuse

Definition

Ontology Design Patterns (ODPs) Reuse is defined as the activity of using available ontology design patterns in the solution of different modeling problems during the development of new ontologies.

Goal

The goal is to allow the reuse of ODPs during the ontology development in order to facilitate the solution of modeling issues and to improve interoperability.

Input

Requirements from the Ontology Requirements Specification Document.

Output

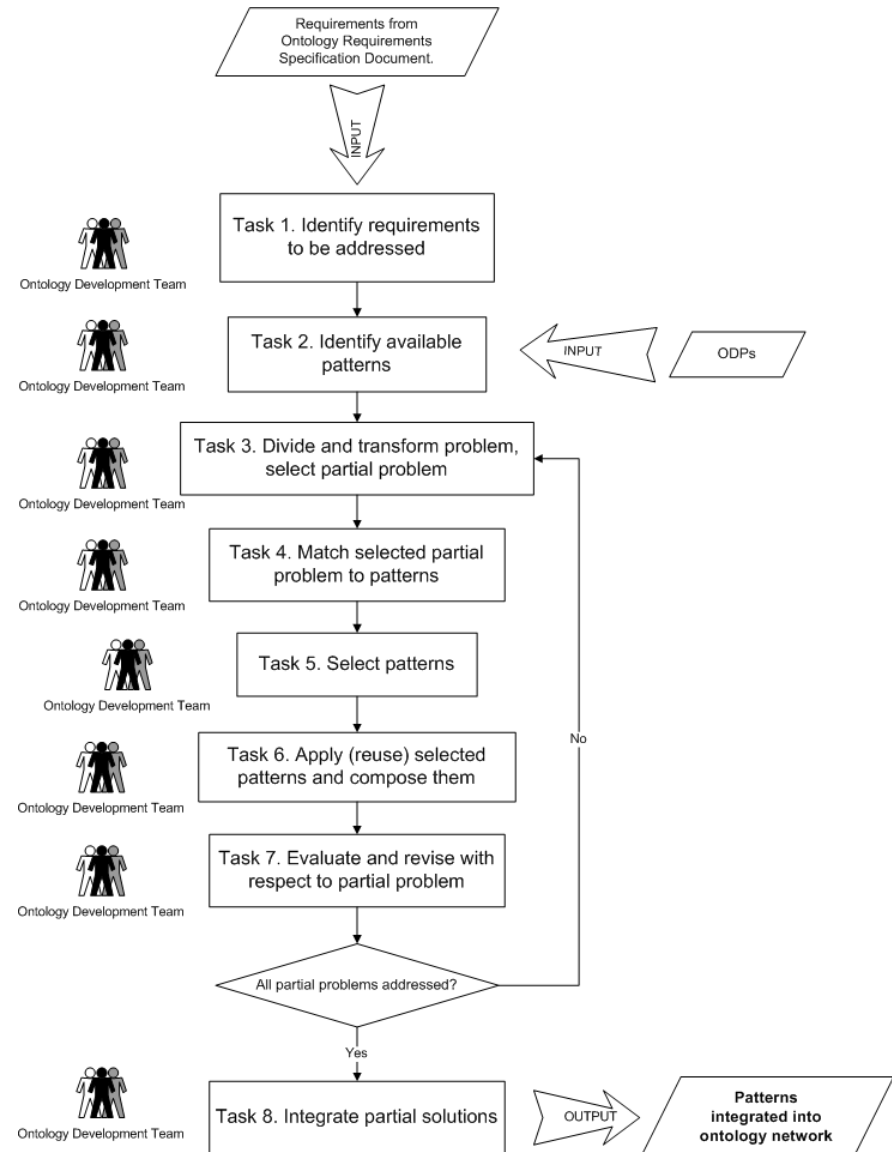
Ontology design patterns integrated into the ontology network being developed.

Who

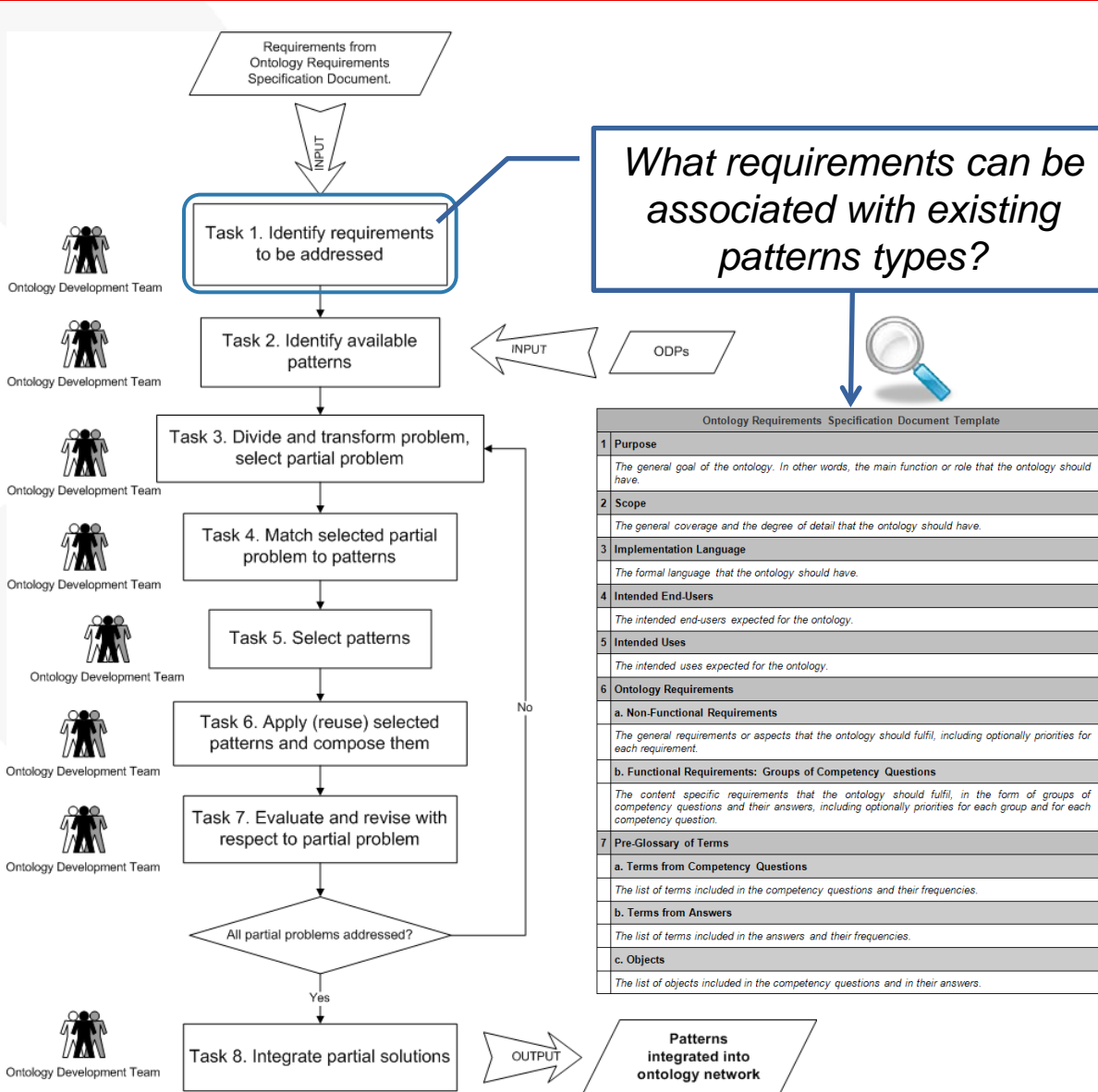
The ontology development team.

When

During the development of the Ontology Conceptualization activity, the Ontology Formalization activity, and/or the Ontology Implementation activity.

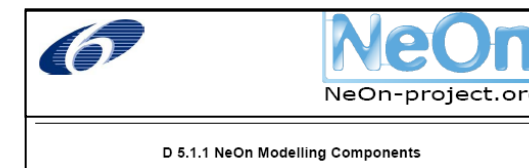
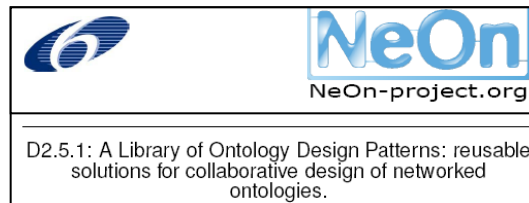
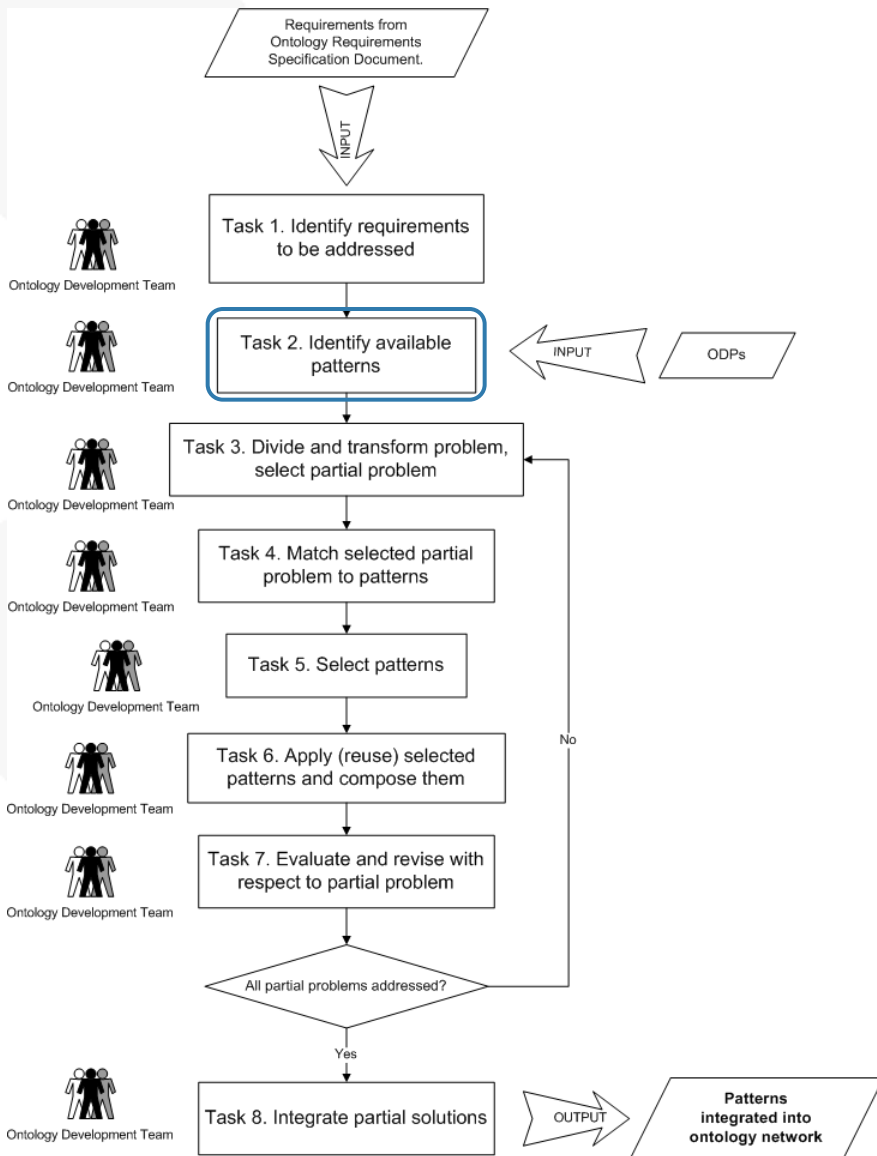


ODPs Reuse: Methodological Guidelines (II)

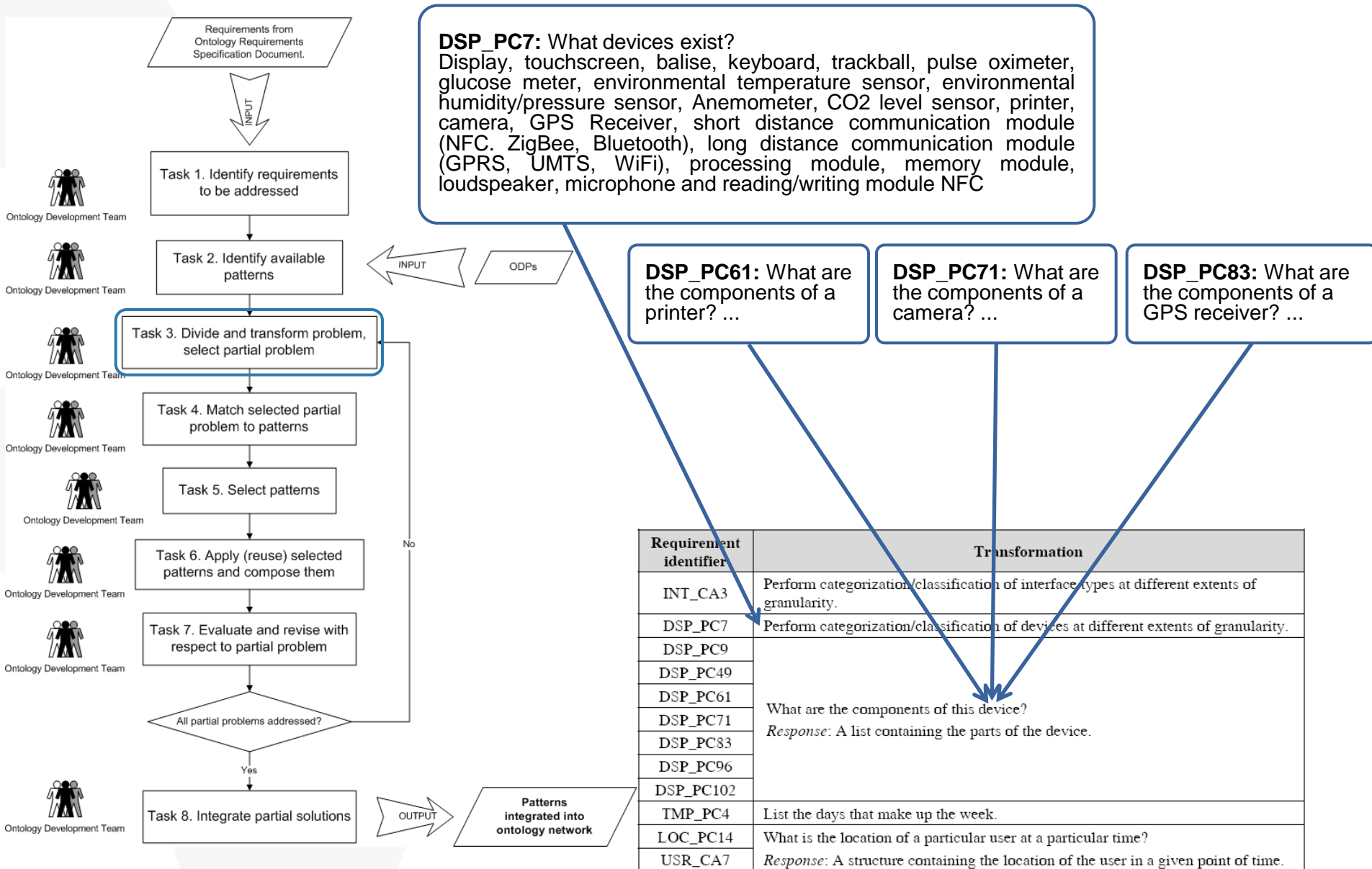


Non-functional requirements	
<ul style="list-style-type: none"> The ontology should be modular. 	
Functional requirements	
Requirement identifier	Requirement
CQ identifier	CQ and its response
DSP_PC7	<p>What devices exist?</p> <p>Display, touchscreen, balise, keyboard, trackball, pulse oximeter, glucose meter, environmental temperature sensor, environmental humidity/pressure sensor, Anemometer, CO2 level sensor, printer, camera, GPS Receiver, short distance communication module (NFC, ZigBee, Bluetooth), long distance communication module (GPRS, UMTS, WiFi), processing module, memory module, loudspeaker, microphone and reading/writing module NFC</p>
DSP_PC9	<p>What are the components of a display?</p> <p>A display is composed by:</p> <ul style="list-style-type: none"> - input interfaces - presentation surface - control interfaces - power system
DSP_PC49	<p>What are the components of a CO2 level sensor?</p> <p>A CO2 level sensor is composed by:</p> <ul style="list-style-type: none"> - source - power system - detector - amplifier - output /download data port (optional)
DSP_PC61	<p>What are the components of a printer?</p> <p>A printer is composed by:</p> <ul style="list-style-type: none"> - storage device - leaf storage receptacle - printhead - ink container - processing device - communication interface - screen
DSP_PC71	<p>What are the components of a camera?</p> <p>A camera is composed by:</p> <ul style="list-style-type: none"> - lens - image capture device - image processing device - storage device - communication interface - positioning device - lighting device - screen - microphone
DSP_PC83	<p>What are the components of a GPS receiver?</p> <p>A GPS receiver is composed by:</p> <ul style="list-style-type: none"> - antenna - signal processor - processing module - communication interface - screen - speaker - microphone
DSP_PC96	<p>What are the components of a speaker?</p> <p>A speaker is composed by:</p> <ul style="list-style-type: none"> - communication interface - amplifier - decoder - signal processing module - active element - casing
DSP_PC102	<p>What are the components of a microphone?</p> <p>A microphone is composed by:</p> <ul style="list-style-type: none"> - diaphragm - coil - permanent magnet - A condenser microphone is composed by: - diaphragm of lightweight and flexible membrane - rigid backplate - cable to the preamplifier - bias voltage feeder
TMP_PC4	<p>What are the days of the week?</p> <p>Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday</p>
LOC_PC14	<p>When was the last time that the user was in the location X?</p> <p>The day Z from xx: xx to yy: yy</p>
NL sentence identifier	Affirmative NL Sentence
INT_CA3	<p>The interfaces types are:</p> <ul style="list-style-type: none"> - conversational - gestural - graphic - natural language - command line - multi screen - touch - textual - vocal - web
USR_CA7	<p>A user will be in a specific location at any given time. The possible physical movement of the user is associated with this aspect</p>

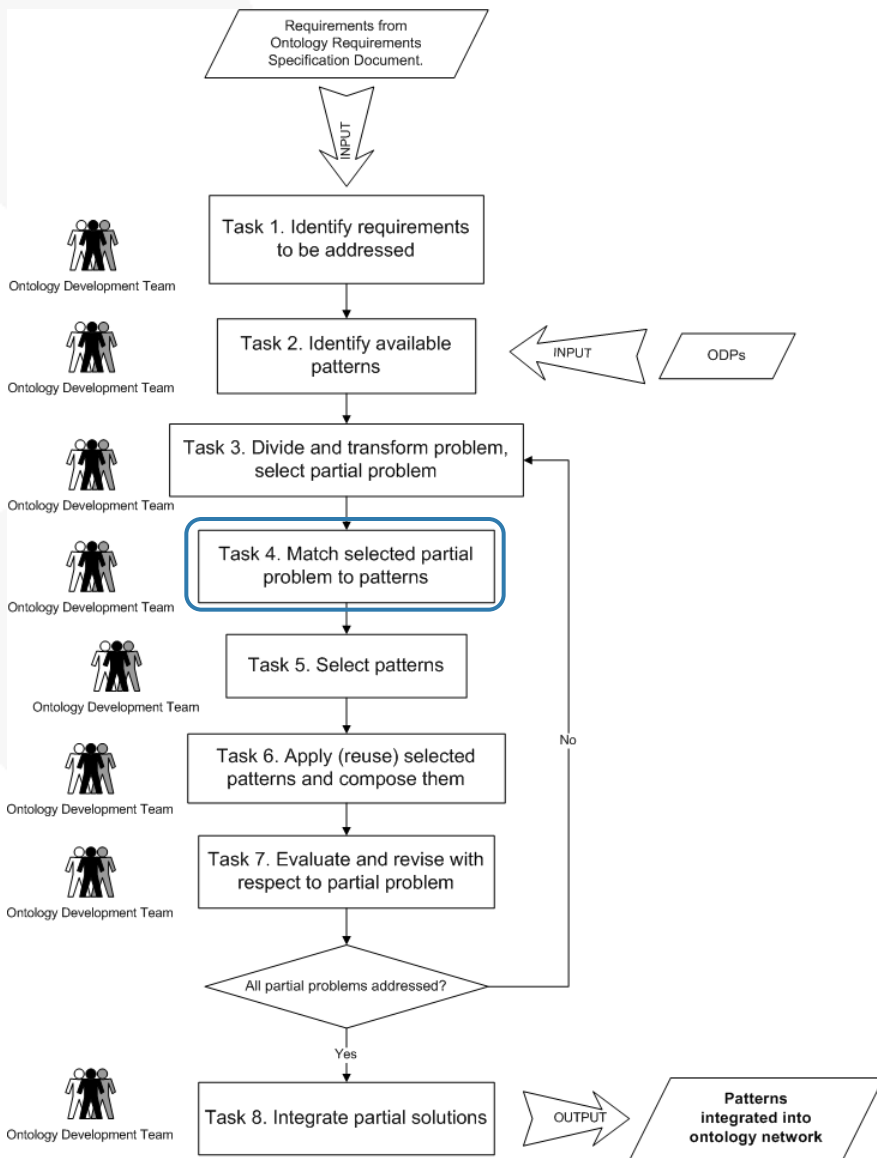
ODPs Reuse: Methodological Guidelines (III)



ODPs Reuse: Methodological Guidelines (IV)



ODPs Reuse: Methodological Guidelines (V)



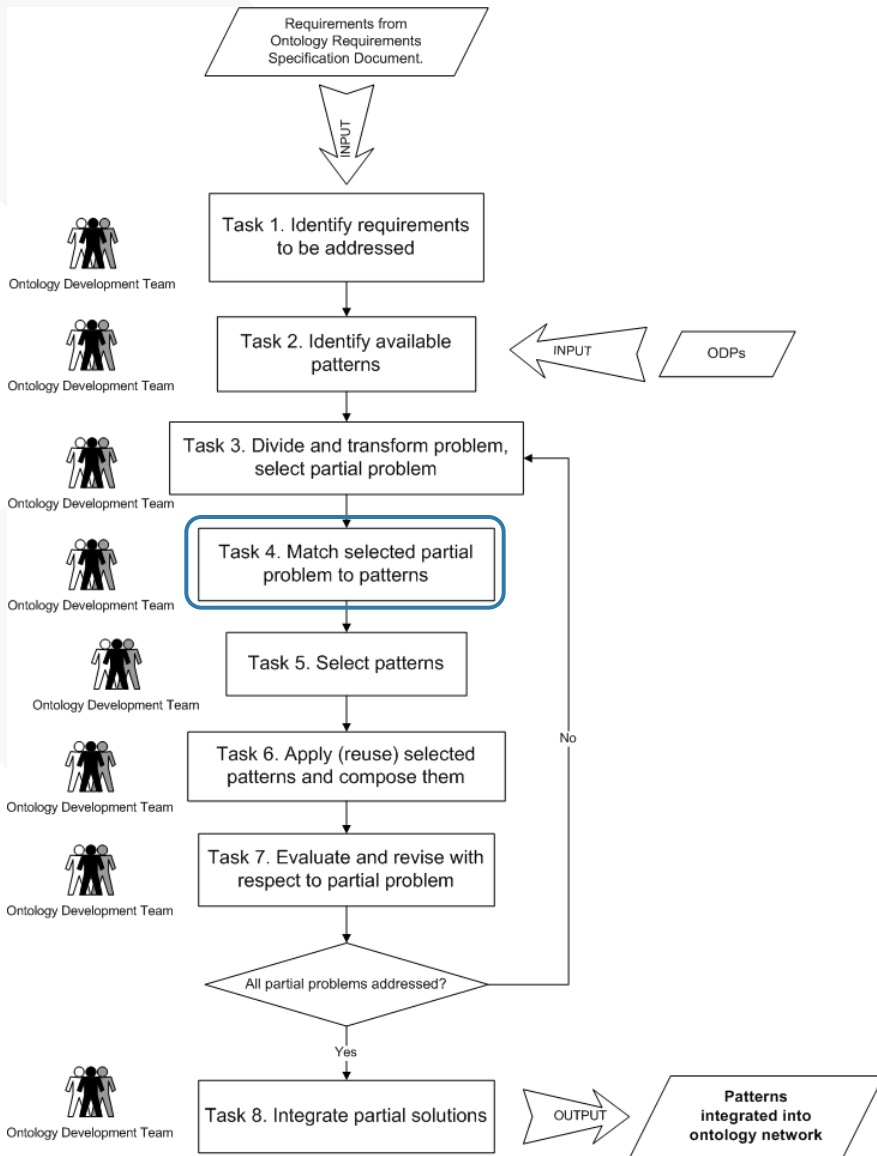
Some heuristics

If a requirement is about *types and subtypes of a given concept*,
then the “**Taxonomy (AP-TX-01)**” pattern could be reused

If a requirement is about *parts of something*,
then the “**Componentency**”, “**Componentency (CP-COM-01)**” or “**Part-Whole Relation (CP-PW-01)**” patterns could be reused

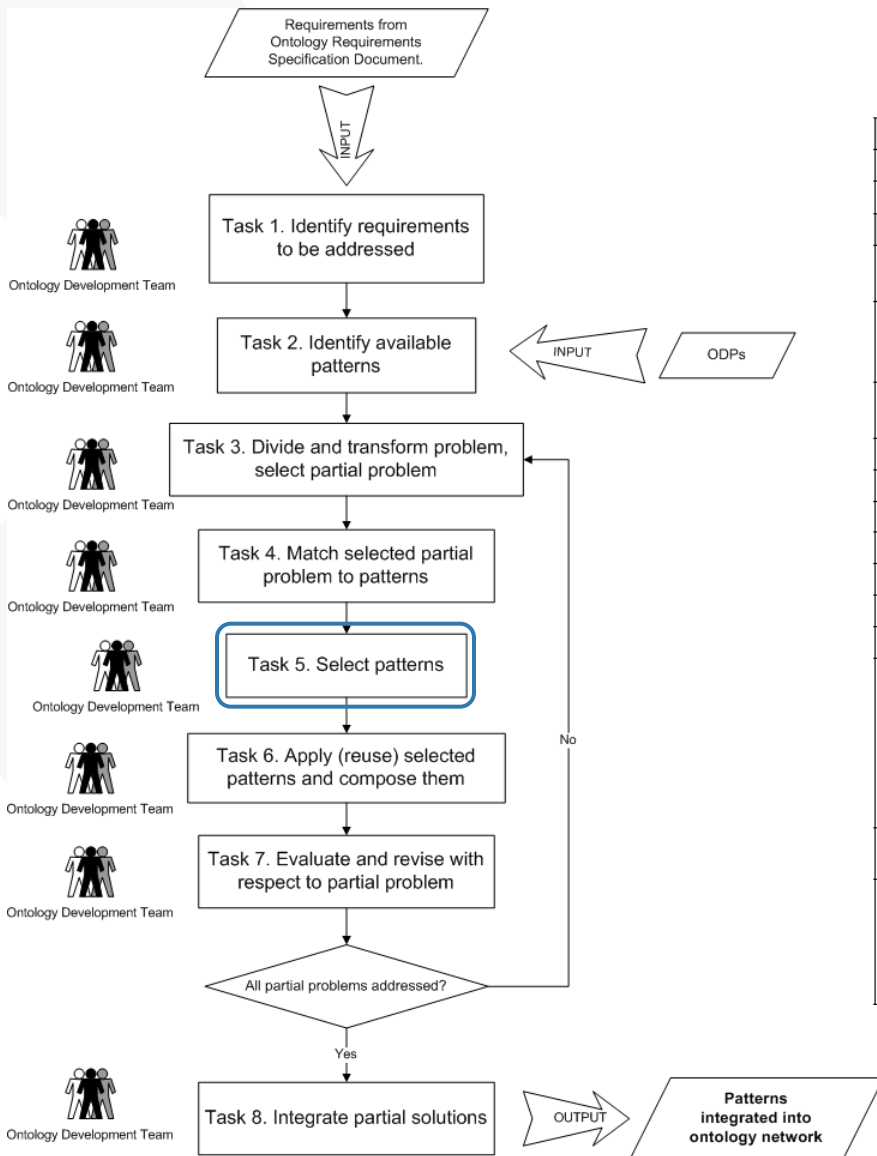
If a requirement mentions *something about a list of values*,
then the “**Specified Values in OWL: "value partitions" and "value sets"**”, the “**Specified Values: Set of Individuals (LP-SV-01)**” or the “**Specified Values: Subclasses (LP-SV-02)**” patterns could be reused.

ODPs Reuse: Methodological Guidelines (V)



Non-functional requirements		
Requirement		Suitable pattern(s)
<ul style="list-style-type: none"> The ontology should be modular. 		<ul style="list-style-type: none"> Modular Architecture (AP-MD-01)
Functional requirements		
Requirement identifier	Transformation	Suitable pattern(s)
INT_CA3	Perform categorization/classification of interface types at different extents of granularity.	<ul style="list-style-type: none"> Taxonomy (AP-TX-01)
DSP_PC7	Perform categorization/classification of devices at different extents of granularity.	<ul style="list-style-type: none"> Taxonomy (AP-TX-01)
DSP_PC9	What are the components of this device?	<ul style="list-style-type: none"> Componenty Componenty (CP-COM-01) Part-Whole Relation (CP-PW-01)
DSP_PC49		
DSP_PC61		
DSP_PC71		
DSP_PC83		
DSP_PC96		
DSP_PC102		
TMP_PC4	List the days that make up the week.	<ul style="list-style-type: none"> Specified Values in OWL: "value partitions" and "value sets" Specified Values: Set of Individuals (LP-SV-01) Specified Values: Subclasses (LP-SV-02)
LOC_PC14	What is the location of a particular user at a particular time?	<ul style="list-style-type: none"> N-Ary Participation N-ary Relations. Pattern 1: Introducing a new class for a relation N-ary Participation (CP-NPAR-01) N-ary Relation: New Class (LP-NR-01)
USR_CA7		

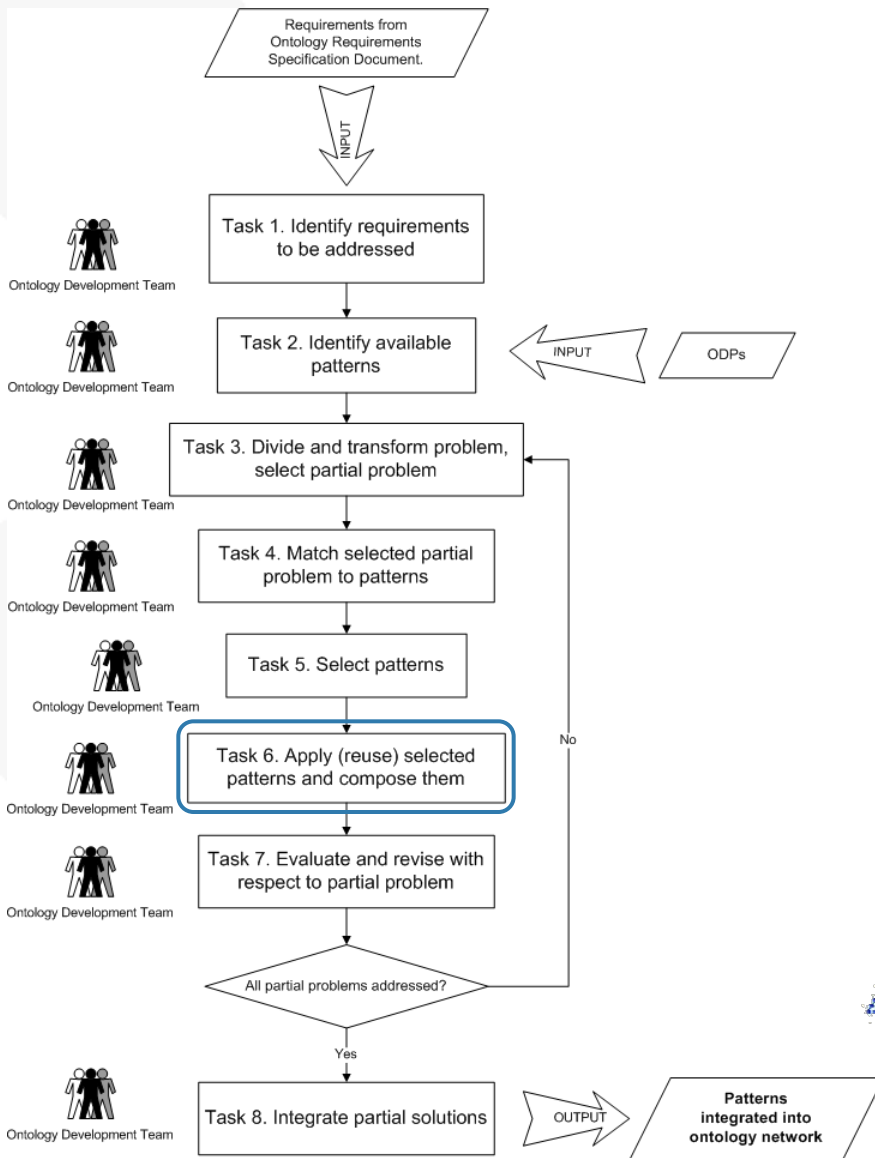
ODPs Reuse: Methodological Guidelines (VI)



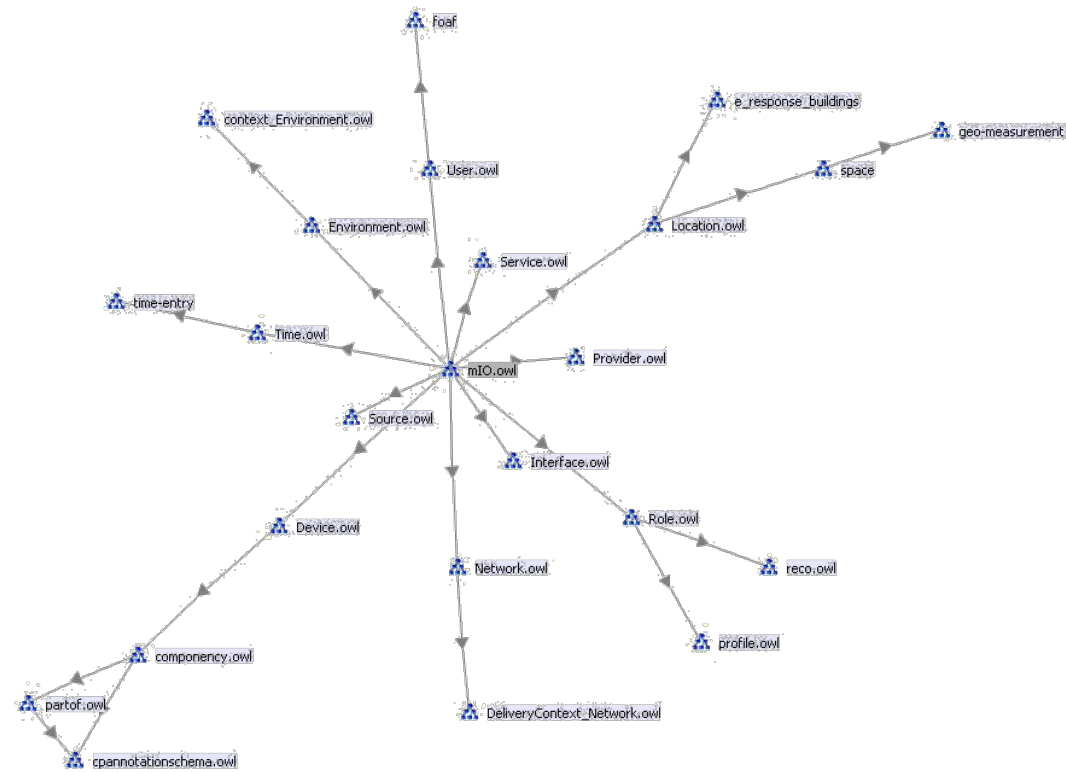
Non-functional requirements		
Requirement		Suitable pattern(s)
• The ontology should be modular.		• Modular Architecture (AP-MD-01) ✓
Functional requirements		
Requirement identifier	Transformation	Suitable pattern(s)
INT_CA3	Perform categorization/classification of interface types at different extents of granularity.	• Taxonomy (AP-TX-01) ✓
DSP_PC7	Perform categorization/classification of devices at different extents of granularity.	• Taxonomy (AP-TX-01) ✓
DSP_PC9	What are the components of this device?	• Componency ✓ • Componency (CP-COM-01) ✓ • Part-Whole Relation (CP-PW-01) ✗
DSP_PC49		
DSP_PC61		
DSP_PC71		
DSP_PC83		
DSP_PC96		
DSP_PC102		
TMP_PC4	List the days that make up the week.	• Specified Values in OWL: "value partitions" and "value sets" ✓ • Specified Values: Set of Individuals (LP-SV-01) ✓ • Specified Values: Subclasses (LP-SV-02) ✗
LOC_PC14	What is the location of a particular user at a particular time?	• N-Ary Participation ✗ • N-ary Relations. Pattern 1: Introducing a new class for a relation ✓
USR_CA7		• N-ary Participation (CP-NPAR-01) ✗ • N-ary Relation: New Class (LP-NR-01) ✓

ODPs Reuse: Methodological Guidelines (VII)

During the **conceptualization** activity we have reused the “Modular Architecture (AP-MD-01)” pattern to define the architecture of the ontology network.



Modular Architecture

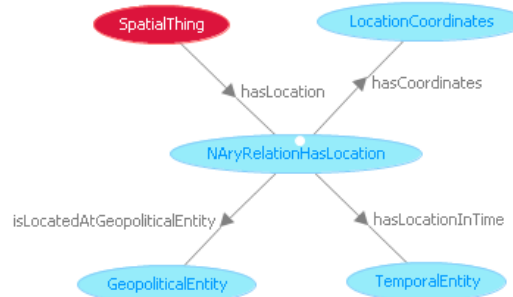


ODPs Reuse: Methodological Guidelines (VIII)

During the **implementation** activity we have reused the “N-ary Relation: New Class (LP-NR-01)” pattern to

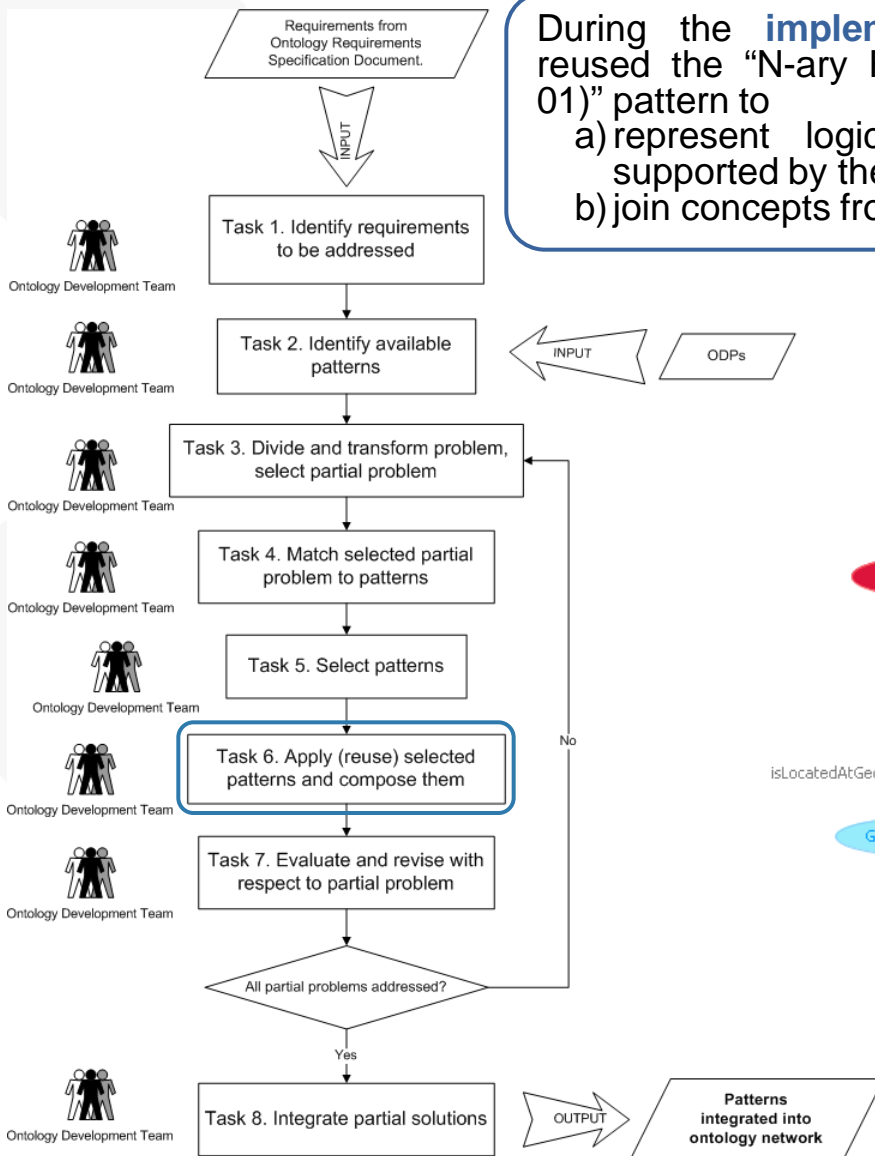
- a) represent logical structures that are not supported by the ontology language
- b) join concepts from different ontologies

N-Ary Relation: New Class



Taxonomy

- Device
 - Anemometer
 - Balise
 - Camera
 - CO2LevelSensor
- Display
 - CRTDisplay
 - LaserDisplay
 - LCDDisplay
 - OLEDDisplay
 - PlasmaDisplay
 - SEDDisplay
 - TFTDisplay
- EnvironmentalHumidity/PressureSensor
- EnvironmentalTemperatureSensor
- GlucoseMeter
- GPSReceiver
- Keyboard
 - CapacitiveKeyboard
 - MembraneKeyboard
 - MetallicContactKeyboard
 - PiezoelectricKeyboard
 - RubberDomeKeyboard
- LargeDistanceCommunicationModule
 - GPRSLargeDistanceCommunicationModule
 - UMTSLargeDistanceCommunicationModule
 - WiFiLargeDistanceCommunicationModule
- Loudspeaker
- MemoryModule
- Microphone
- Printer
- ProcessingModule
- PulseOximeter
- Readin/WritingModuleNFC
- ShortDistanceCommunicationModule
 - BlueToothShortDistanceCommunicationModule
 - NFCShortDistanceCommunicationModule
 - ZigBeeShortDistanceCommunicationModule
- TouchScreen
 - AcousticPulserecognitionTouchScreen
 - AcousticWaveTouchScreen
 - CapacitiveTouchScreen
 - InfraredTouchScreen
 - OpticalImagingTouchScreen
 - PressureSensorTouchScreen
 - ResistiveTouchScreen
- Trackball

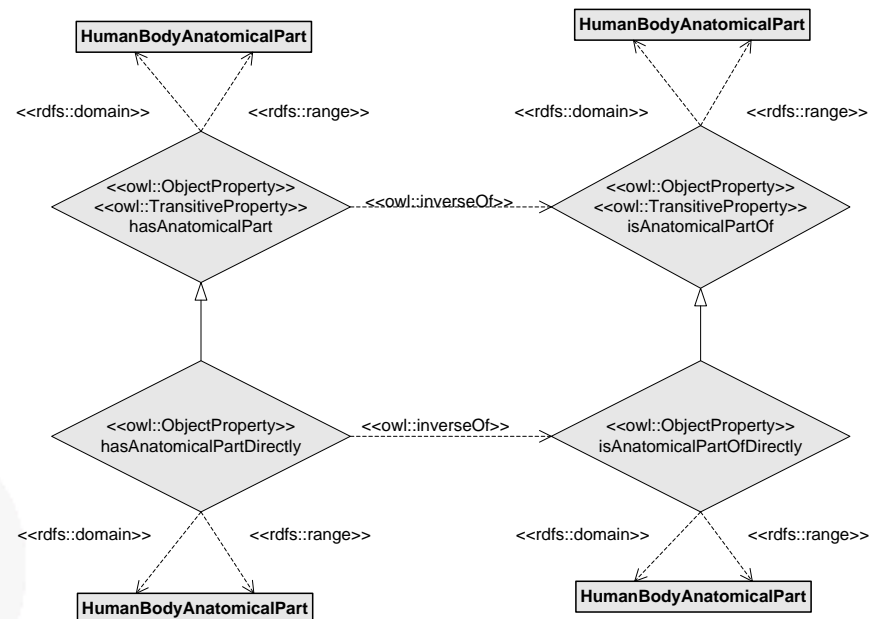




We are going to **model** the following problem within the Anatomy Ontology:

- The Knee is part of the Leg
- The Front of the Knee is part of the Knee
- The Front of the Knee also forms part of the Leg
- The same for other parts of the body

The **Content Pattern for modelling a Part-Whole Relation** (CP-PW-01) was reused to structure the different parts of the body and the relation among them.





Methodological Guidelines for Reusing Ontology Design Patterns

Mari Carmen Suárez-Figueroa