ISP-OPL – ISO-based Software Process OPL

An Ontology Pattern Language for modeling Software Processes in the context of ISO SC7.

An Ontology Pattern Language (OPL) [1] aims to provide holistic support for using Domain-related Ontology Patterns (DROPs) in ontology development for a specific application domain. It provides explicit guidance on what problems can arise in that domain, informs the order to address these problems, and suggests one or more patterns to solve each specific problem. Moreover, an OPL supports the explicit consideration of complementing or conflicting pattern combinations to resolve a given problem, along with guidelines for integrating patterns into a concrete ontology conceptual model.

An OPL indicates explicitly which referenced patterns address mandatory aspects and which ones address optional aspects. To ensure a stable and sound pattern application, the patterns are presented in the suggested application order. OPLs are structured to support and encourage the application of one pattern at a time, in the order defined by the pattern sequences that result from the chosen paths through the language process. This guideline ensures that the main property of piecemeal growth is preserved: the 'whole' always precedes its 'parts'.

The ISP-OPL addresses three main aspects dealt by ISO software process standards: Work Units, including patterns to define the composition of work units, dependence between work units, and scheduling of work units, among others; Work Products, considering the nature of work products produced during the software process and how they are handled in work units; and Human Resources, dealing with how people are organized in teams, allocated to tasks and perform work units. The patterns composing the language were based on an Ontological Analysis of ISO/IEC 24744 [2] and some of them imported and adapted from SP-OPL [1] and E-OPL [3].

Table 1 presents the patterns that compose the ISP-OPL, grouped in each one of these problem areas. Figure 1 presents the OPL Process, with the recommended application paths. In the sequel, each pattern is presented in detail.

ISP-OPL Patterns

Table 1. Domain-Related Ontology Patterns in the ISP-OLP.

ID	Name	Intent
Work Units		
WUD	Work Unit Dependence	Represents the dependences between work units.
WUC	Work Unit Composition	Establishes the composition of work units in terms of processes and tasks.
PPD	Project Process Definition	Defines the project of a defined process.
WUS	Work Unit Scheduling	Defines the time boundary for work units.
PWUT	Performed Work Unit Tracking	Register the tracking between performed and scheduled work units.
PWUD	Performed Work Unit Dependence	Represents the dependences between performed work units.
PWUC	Performed Work Unit Composition	Establishes the composition of work units in terms of performed processes and tasks.
PPP	Project Process Performing	Defines the project of a performed process.
Human Resources		
HRE	Human Resource Employment[1]	Establishes the human resources employed in organizations.
PTD	Project Team Definition[1]	Defines project teams.
OTD	Organizational Team Definition[1]	Defines organizational teams.
TRD	Team Role Definition[1]	Sets the roles to be performed by teams.
TMR	Team Membership with Role[1]	Establishes the members of a team, defining the period of time of such membership and the role to be played by a team member.
TMs	Team Membership simplified[1]	Establishes the members of a team.
RPL	Role Planning	Represents the roles that should perform a work unit.
StD	Stakeholder Definition	Defines the stakeholders (people or teams) involved in a project.
StA	Stakeholder Allocation	Establishes the allocation of stakeholders to scheduled work units, defining a period of time and the role to be played by the stakeholder in such allocation.
StAs	Stakeholder Allocation simplified	Establishes the allocation of stakeholders to scheduled work units.
PPa	Producer Participation	Registers the stakeholders participation, during a period of time, in performed work units.
PPas	Producer Participation simplified	Registers the stakeholders participation in performed work units.
Work Products		
WPN	Work Product Nature	Represents the different types of work products, according to their nature.
WPC	Work Product Composition	Distinguishes between composite and simple work products, representing the decomposition of composite work products.
DocD	Document Depiction	Defines the work products that a document depicts.
WPP	Work Product Participation	Establishes the participation (creation, change or usage) of work products in performed work units.
WPCrea	Work Product Creation	Establishes which work products are created by a work unit.
WPUse	Work Product Usage	Establishes which work products are used in a work unit.
WPChan	Work Product Change	Establishes which work products are changed in a work unit.

[1] Adapted from E-OPL

ISP-OPL Process

Figure 1 presents a UML activity diagram showing the language paths of ISP-OPL. In the diagram, Domain-Related Ontology Patterns (DROPs) are represented by action nodes (the labeled rounded rectangles). Initial nodes (solid circles) represent entry points in the OPL, i.e., DROPs in the language that can be used without solving other problems first. Control flows (arrowed lines) represent the admissible sequences in which DROPs can be used. Merge points (diamond-shaped symbols) represent the merge of paths in the OPL. Join/Fork nodes (line segments) represent the conjunction of paths (join) or independent and possibly parallel paths (fork). Finally, an extension to the original UML notation (dotted lines with arrows) is used to represent variant patterns, i.e. patterns that can be used to solve the same problem in different ways.

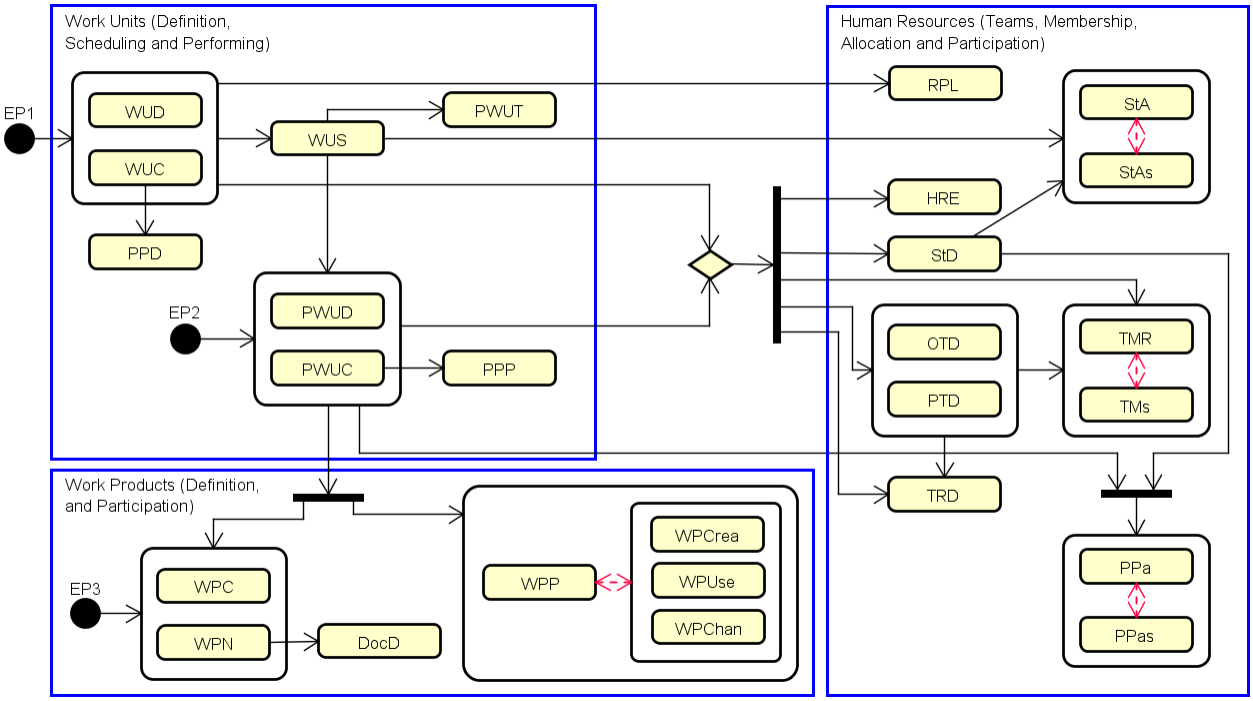


Figure 1. ISP-OPL Process

ISP-OPL Entry Points

ISP-OPL has three entry points. The ontology engineer should choose one of them, depending on the scope of the specific software process ontology being developed.

EP1 – This entry point should be chosen when the requirements for the new ontology include the definition and planning of work units.

EP2 – This entry point should be chosen when the requirements for the new ontology consider only the execution of work units.

EP3 – This entry point should be chosen when the requirements for the new ontology consider only structuring work products.

If the ontology engineer needs to model other related areas, other OPLs could be used, such as SP-OPL [1], for Standard Processes, and E-OPL [3] for Organizational Structure and Human Resources.

Work Units

WUD – Work Unit Dependence

Name: Work Unit Dependence (WUD)

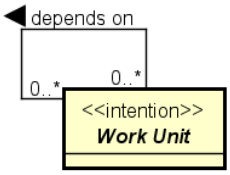
Intent: To represent the dependencies between work units.

Rationale: A *Work Unit* can depend on other *work units*. There are several situations in which dependencies can occur, such as a *work unit* that uses the work products created by another or a *work unit* that needs a resource being used by another.

Competency Questions:

- From which work unit does a work unit depend on to be performed?

Conceptual Model



Axiomatization:

A1	$\forall$a,b,c: (dependsOn(a, b) $\wedge$ dependsOn(b, c)) $\rightarrow$ dependsOn(a, c)
A2	$\forall$a,b: dependsOn(a, b) $\rightarrow$ (a != b)
A3	$\forall$a,b: dependsOn(a, b) $\rightarrow \neg$dependsOn(b, a)

- Axiom A1: If a **Work Unit** *a* depends on a **Work Unit** *b* and *b* depends on a **Work Unit** *c*, then *a* depends on *c* (transitivity).
- Axiom A2: If a **Work Unit** *a* depends on a **Work Unit** *b*, then *a* is different from *b* (non-reflexibility).
- Axiom A3: If a **Work Unit** *a* depends on a **Work Unit** *b*, then *b* does not depend on *a* (asymmetry).

WUC – Work Unit Composition

Name: Work Unit Composition (WUC)

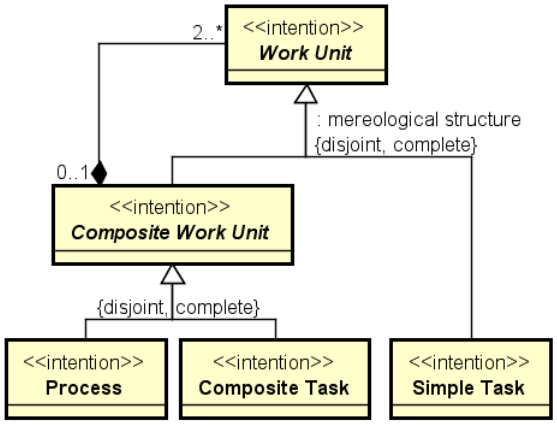
Intent: To represent the composition of work units in terms of other work units.

Rationale: *Work Units* can be composed of other *work units*. Mereologically, a *work unit* is simple or composed of two or more parts. At the basic level, there are *Simple Tasks* that compose other *work units*, but are not decomposed. *Composite Tasks*, in turn, are composed of other tasks (*composite* or *simple tasks*). At the higher level, *Processes* are also composed of tasks, but do not compose other *work units*.

Competency Questions:

- Concerning their mereological structure, what are the possible types of work units?
- How a work unit is composed of work units?

Conceptual Model



Axiomatization:

A1	$\forall w,c\ partOf(w,c) \rightarrow (w \mathrel{!}= c)$
A2	$\forall p\ Process(p) \rightarrow \neg\exists w\ WorkUnit(w) \wedge partOf(p,w)$

- Axiom A1: A **Work Unit** cannot be part of itself.
- Axiom A2: A **Process** cannot be part of any **Work Unit**.

PPD – Project Process Definition

Name: Project Process Definition (PPD)

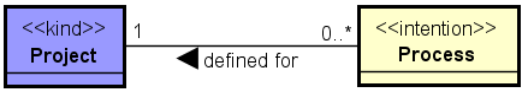
Intent: To represent the project to which a process is defined.

Rationale: A *Process* is defined for a *Project*. In the context of a *project*, many *processes* can be defined, for different purposes.

Competency Questions:

- For which project is a process defined? What are the processes of a project?

Conceptual Model



Axiomatization:

-

WUS – Work Unit Scheduling

Name: Work Unit Scheduling (WUS)

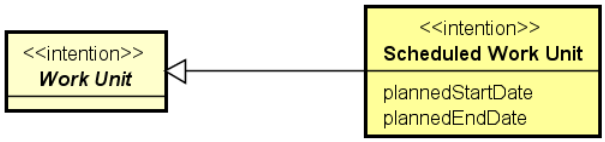
Intent: To represent the time boundary for work units.

Rationale: A *Work Unit*, once defined, can be *Scheduled* in order to be performed in a certain period of time, establishing planned start and end dates.

Competency Questions:

- For what period is a work unit planned?

Conceptual Model



Axiomatization:

A1	∀a: ScheduledWorkUnit(a) → (a.plannedStartDate < a.plannedEndDate)

- Axiom A1: For a **Scheduled Work Unit**, ***plannedStartDate*** should occur before ***plannedEndDate***.

PWUT – Performed Work Unit Tracking

Name: Performed Work Unit Tracking (PWUT)

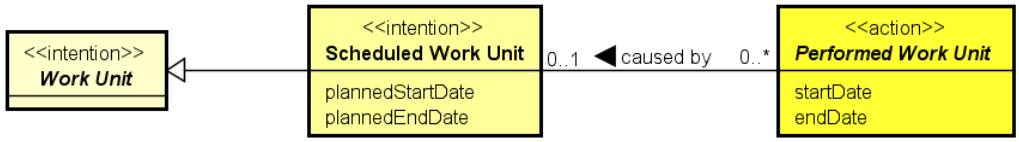
Intent: To represent the tracking between performed and scheduled work units.

Rationale: *Performed Work Units* are actions caused by *Scheduled Work Units*. A *Scheduled Work Unit*, once planned, can be performed.

Competency Questions:

- When a work unit was performed?
- Which scheduled work unit has caused the execution of a performed work unit?

Conceptual Model



Axiomatization:

A1	$\forall a$: PerformedWorkUnit(a) $\rightarrow$ (a.startDate < a.endDate)

- Axiom A1: For a **Performed Work Unit**, ***startDate*** occurs before ***endDate***.

PWUD – Performed Work Unit Dependence

Name: Performed Work Unit Dependence (PWUD)

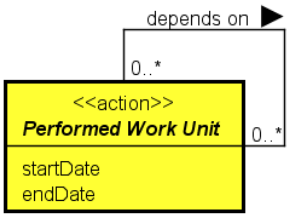
Intent: To represent the dependencies between performed work units.

Rationale: For a *Performed Work Unit* (an actual occurrence of an action) to be accomplished, other *performed work units* may have to be accomplished first, indicating that there is a dependency between them. There are several situations in which dependencies can occur, such as a *performed work unit* that has used work products created by another *performed work unit*, or a *performed work unit* that has needed a resource which were being used by another *performed work unit*.

Competency Questions:

- When a work unit was performed?
- From which work unit did a work unit depend on to be performed?

Conceptual Model



Axiomatization:

A1	∀a: PerformedWorkUnit(a) → (a.startDate < a.endDate)
A2	∀a,b,c: (dependsOn(a, b) ∧ dependsOn(b, c)) → dependsOn(a, c)
A3	∀a,b: dependsOn(a, b) → (a != b)
A4	∀a,b: dependsOn(a, b) → ¬dependsOn(b, a)
A5	∀a,b: dependsOn(a, b) → (a.startDate > b.startDate)

- Axiom A1: For a **Performed Work Unit**, ***startDate*** occurs before ***endDate***.
- Axiom A2: If a **Performed Work Unit** *a* depends on a **Performed Work Unit** *b* and *b* depends on a **Performed Work Unit** *c*, then *a* depends on *c* (transitivity).
- Axiom A3: If a **Performed Work Unit** *a* depends on a **Performed Work Unit** *b*, then *a* is different from *b* (non-reflexibility).
- Axiom A4: If a **Performed Work Unit** *a* depends on a **Performed Work Unit** *b*, then *b* does not depend on *a* (asymmetry).
- Axiom A5: If a **Performed Work Unit** *a* depends on a **Performed Work Unit** *b*, then *a* starts after *b*'s start date.

PWUC – Performed Work Unit Composition

Name: Performed Work Unit Composition (PWUC)

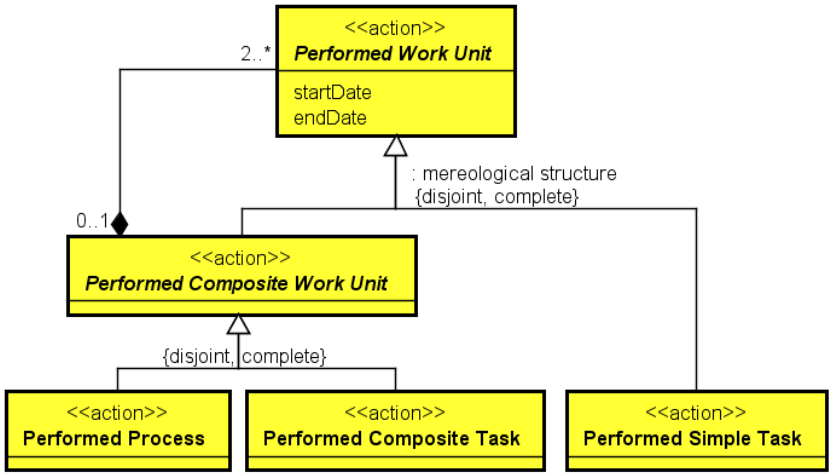
Intent: To represent the composition of performed work units in terms of other performed work units.

Rationale: *Performed Work Units* can be composed of other *performed work units*. Mereologically, a *performed work unit* is simple, or composed of two or more parts. At the basic level, there are *Performed Simple Tasks* that compose other *performed work units*, but are not decomposed. *Performed Composite Tasks*, in turn, are composed of other performed tasks (*composite* or *simple performed tasks*). At the higher level, *Performed Processes* are also composed of performed tasks, but do not compose other *performed work units*.

Competency Questions:

- When a work unit was performed?
- Concerning their mereological structure, what are the possible types of performed work units?
- How a performed work unit is composed of performed work units?

Conceptual Model



Axiomatization:

A1	$\forall$w,c partOf(w,c) $\rightarrow$ (w != c)
A2	$\forall$a: PerformedWorkUnit(a) $\rightarrow$ (a.startDate < a.endDate)
A3	$\forall$p: PerformedProcess(p) $\rightarrow$ $\neg$ $\exists$w PerformedWorkUnit(w) $\wedge$ partOf(p,w)
A4	$\forall$w1,w2: partOf(w2,w1) $\rightarrow$ (w2.startDate >= w1.startDate) $\wedge$ (w2.endDate <= w1.endDate)

- Axiom A1: A **Performed Work Unit** cannot be part of itself.
- Axiom A2: For a **Performed Work Unit**, ***startDate*** occurs before ***endDate***.
- Axiom A3: A **Performed Process** cannot be part of any **Performed Work Unit**.
- Axiom A4: A **Performed Work Unit** that is part of another should occur within the time interval of its whole.

PPP – Project Process Performing

Name: Project Process Performing (PPP)

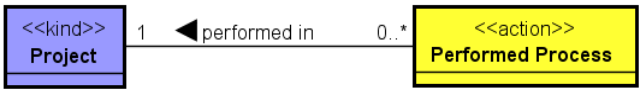
Intent: To represent the project in which context a performed process was executed.

Rationale: A *Performed Process* is a process that was executed in the context of a *Project*. Many processes can be performed in the context of a *project*, for different purposes.

Competency Questions:

- In the context of which project was a process performed?

Conceptual Model



Axiomatization:

-

Human Resources

HRE – Human Resource Employment (based on E-OPL)

Name: Human Resource Employment (HRE)

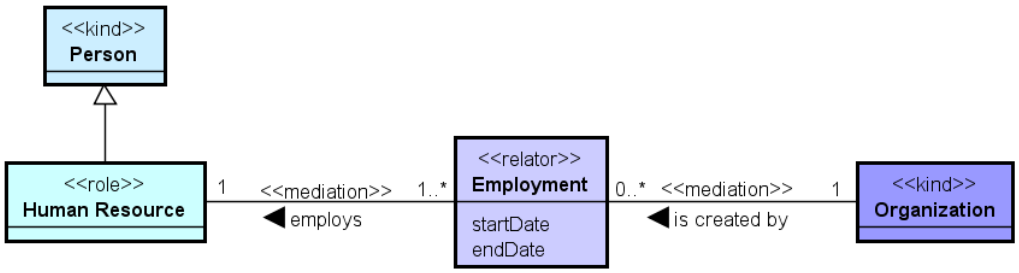
Intent: To represent the human resources employed in an organization.

Rationale: An *Organization* can employ people as *Human Resources*. When employed, a *Person* assumes the *Human Resource* role for the period of time defined by the *employment* relator.

Competency Questions:

- Who are the human resources employed in an organization?
- What is the employment period?

Conceptual Model



Axiomatization:

A1	∀a: Employment(a) → (a.startDate < a.endDate)

- Axiom A1: For an **Employment**, ***startDate*** occurs before ***endDate***.

PTD – Project Team Definition (based on E-OPL)

Name: Project Team Definition (PTD)

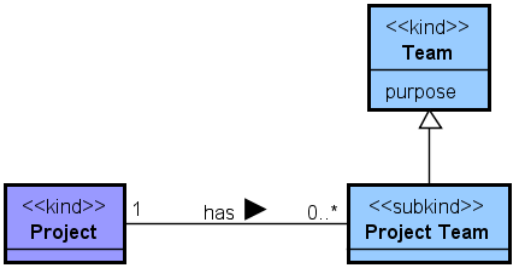
Intent: To represent project teams, i.e. teams that act only in the context of a given project.

Rationale: A *Project* can define *Project Teams* to serve to a defined purpose in the *project*.

Competency Questions:

- What are the Teams of a Project?

Conceptual Model



Axiomatization:

-

OTD – Organizational Team Definition (based on E-OPL)

Name: Organizational Team Definition (OTD)

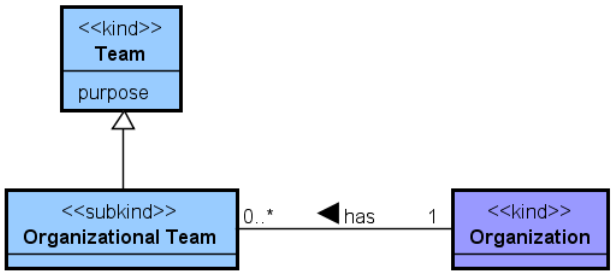
Intent: To represent organizational teams, i.e. teams that act in different projects of a given organization.

Rationale: An *Organization* can define *Organizational Teams* to act, according to its purpose, in different projects of the *organization*.

Competency Questions:

- What are the Organizational Teams of an Organization?

Conceptual Model



Axiomatization:

-

TRD – Team Role Definition (based on E-OPL)

Name: Team Role Definition (TRD)

Intent: To represent the roles that a team can play.

Rationale: A *Team* can perform several *Roles* defined by the organization to reach its purposes.

Competency Questions:

- What Roles does a Team perform?

Conceptual Model

Axiomatization:

-

TMR – Team Membership with Role (based on E-OPL)

Name: Team Membership with Role (TMR)

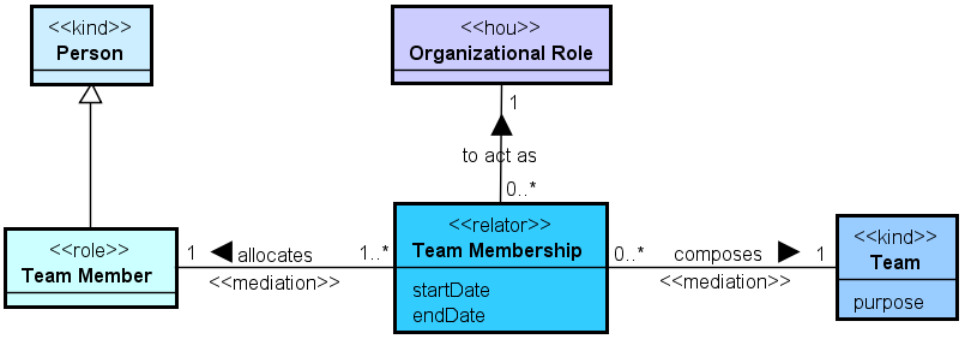
Intent: To represent the members of a team, defining the period of time of such membership, and the role to be played by the team member in this team.

Rationale: A *Person* can be allocated as a member of a *Team*, acting according to a defined *Organizational Role*. Thus, the *Team Membership* relator defines which *Team Member* is allocated to which *Team*, and the role he/she plays in that team, as well as the period of time of that membership.

Competency Questions:

- Who are the Members of a Team?
- Which Organizational Role does a Team Member assume in the context of a Team?
- What is the membership period?

Conceptual Model



Axiomatization:

A1	∀a: TeamMembership(a) → (a.startDate < a.endDate)
A2	∀tm,tm',p,p',t,t',r,r': (allocates(tm,p) ∧ composes (tm,t) ∧ actAs(tm,r)) ∧ (allocates(tm',p') ∧ composes (tm',t') ∧ actAs(tm',r')) → (p !=p' ∨ t !=t' ∨ r != r' ∨ (tm.end <= tm'.start) ∨ (tm'.end <= tm.start))

- Axiom A1: For a **Team Membership**, ***startDate*** occurs before ***endDate***.
- Axiom A2: A **Team Member** cannot assume the same **Organizational Role** in the same **Team** at the same ***time***.

TMs – Team Membership simplified (based on E-OPL)

Name: Team Membership simplified (TMs)

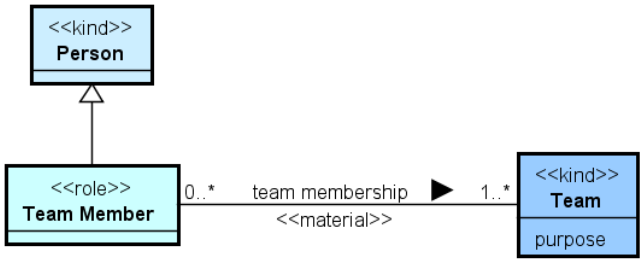
Intent: To represent only the members of a team.

Rationale: A *Person* can be member of a *Team*, becoming a *Team Member*.

Competency Questions:

- Who are the members of a Team?

Conceptual Model



Axiomatization:

-

RPL – Role Planning

Name: Role Planning (RPL)

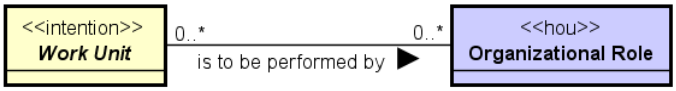
Intent: To represent the roles that should perform a work unit.

Rationale: *Work Units* are to be performed by people playing certain defined *roles*. These *roles* can be planned when the *Work Unit* is defined.

Competency Questions:

- Which are the Organizational Roles planned to perform a Work Unit?

Conceptual Model



Axiomatization:

-

StD – Stakeholder Definition

Name: Stakeholder Definition (StD)

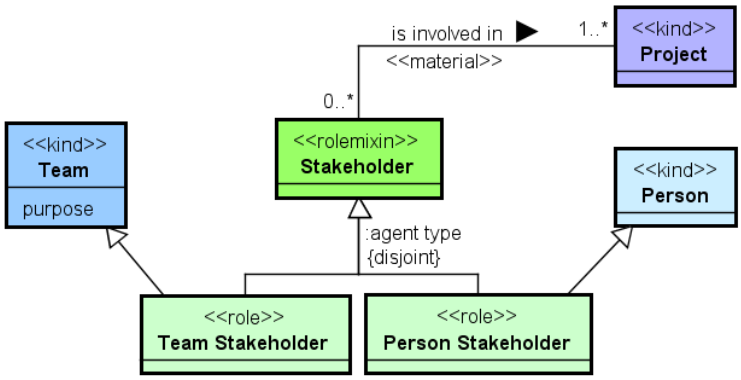
Intent: To represent the stakeholders (people or teams) involved in projects.

Rationale: *Stakeholders* are the agents involved in a *Project*. They can be *Team Stakeholders*, representing *Teams* involved in a *Project*, or *Person Stakeholders*, representing People involved in a *Project*.

Competency Questions:

- Who are the Stakeholders of a Project?
- According to the types of agents, what are the possible types of Stakeholder?

Conceptual Model



Axiomatization:

-

StA – Stakeholder Allocation

Name: Stakeholder Allocation (StA)

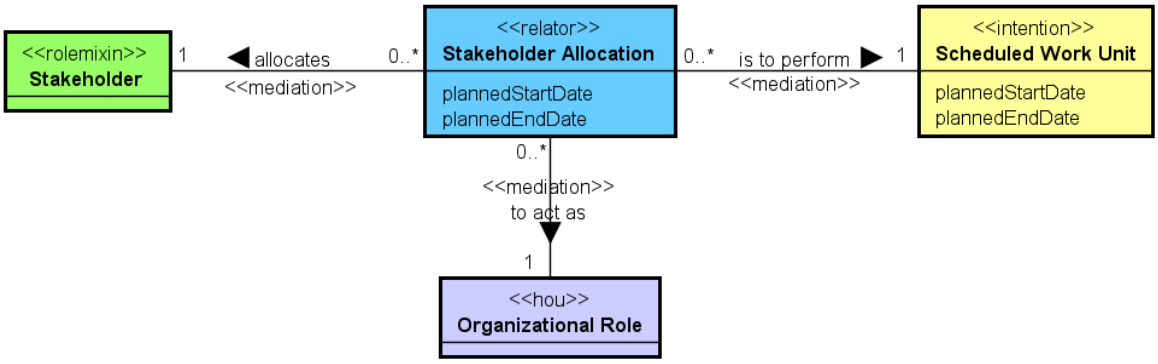
Intent: To represent the allocation of stakeholders in scheduled work units, defining a period of time and the role to be played by the stakeholder in such allocation.

Rationale: A *Stakeholder* can be allocated to perform a *Scheduled Work Unit*, acting according to an *Organizational Role*. Thus, the *Stakeholder Allocation* relator defines which *Stakeholder* is allocated to a given *Scheduled Work Unit*, and the *role* the stakeholder plays in that *allocation*, as well as the allocation period of time.

Competency Questions:

- Who are the Stakeholders allocated to a Scheduled Work Unit?
- Which Organizational Roles does an allocated Stakeholder play in the context of a Scheduled Work Unit?
- What is the allocation period of time?

Conceptual Model



Axiomatization:

A1	∀a: StakeholderAllocation(a) → (a.plannedStartDate < a.plannedEndDate)
A2	∀a,u: isToPerform(a,u) → (a.plannedStartDate >= u.plannedStartDate) ∧ (a.plannedEndDate <= u.plannedEndDate)
A3	∀sa,sa',s,s',wu,wu',r,r': (allocates(sa,s) ∧ isToPerform (sa,wu) ∧ actAs(sa,r)) ∧ (allocates(sa',s') ∧ isToPerform (sa',wu') ∧ actAs(sa',r')) → (s !=s' ∨ wu !=wu' ∨ r != r' ∨ (sa.end <= sa'.start) ∨ (sa'.end <= sa.start))

- Axiom A1: For a **Stakeholder Allocation**, ***plannedStartDate*** occurs before ***plannedEndDate***.
- Axiom A2: The period of a **Stakeholder Allocation** should be within the period of the related **Scheduled Work Unit**.
- Axiom A3: A **Stakeholder** cannot assume the same **Organizational Role** in the same **Scheduled Work Unit** at the same ***time***.

StAs – Stakeholder Allocation simplified

Name: Stakeholder Allocation simplified (StAs)

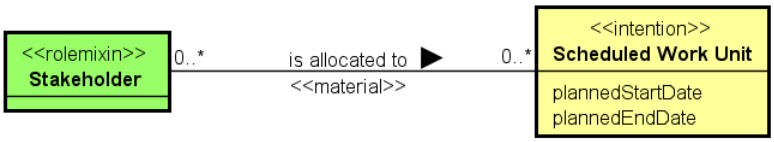
Intent: To represent the allocation of stakeholders to scheduled work units.

Rationale: *Stakeholders*, as involved in a *Project*, can be allocated to *Scheduled Work Units*.

Competency Questions:

- Who are the Stakeholders allocated to a Scheduled Work Unit?

Conceptual Model



Axiomatization:

-

PPa – Producer Participation

Name: Producer Participation (PPa)

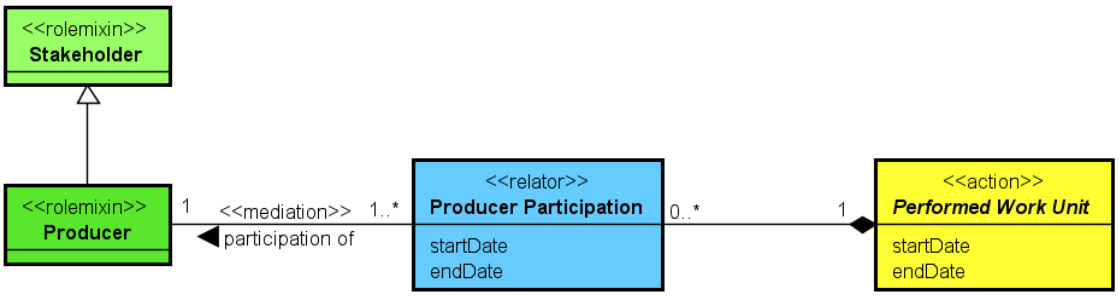
Intent: To represent the stakeholders participating, during a period of time, in a performed work unit.

Rationale: *Stakeholders* participate in *Performed Work Units* during a period of time. In such *participations*, they act as *Producers*.

Competency Questions:

- Who were the Stakeholders that participated in a Performed Work Unit?
- What was the period of time of a producer participation?

Conceptual Model



Axiomatization:

A1	∀a: ProducerParticipation(a) → (a.startDate < a.endDate)
A2	∀p,u: partOf(p,u) → (p.startDate >= u.startDate) ∧ (p.endDate <= u.endDate)

- Axiom A1: For a **Producer Participation**, ***startDate*** occurs before ***endDate***.
- Axiom A2: The period of a **Producer Participation** is within the period of the related **Performed Work Unit**.

PPas – Producer Participation simplified

Name: Producer Participation simplified (PPas)

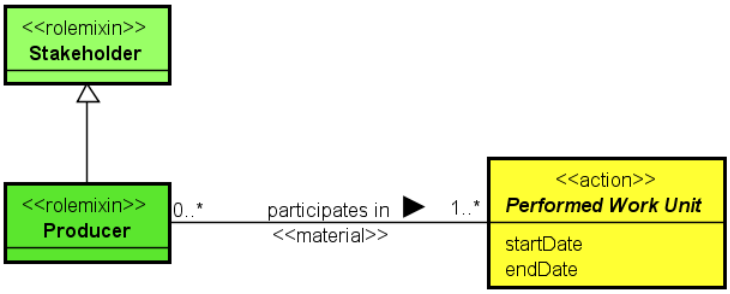
Intent: To represent the stakeholders participating in a performed work unit.

Rationale: *Stakeholders* participate in *Performed Work Units*. In such participations, they act as *Producers*.

Competency Questions:

- Who were the Stakeholders that participated in a Performed Work Unit?

Conceptual Model



Axiomatization:

-

Work Products

WPN – Work Product Nature

Name: Work Product Nature (WPN)

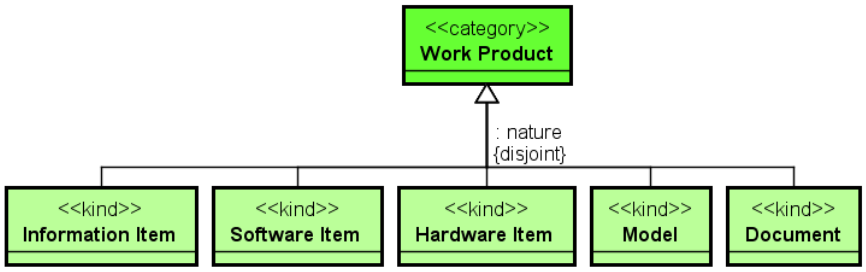
Intent: To represent the different types of work products, according to their nature.

Rationale: *Work Products* can have different natures. The most common are *Document*, *Model*, *Software Item*, *Information Item*, and *Hardware Item*.

Competency Questions:

- According to its nature, what is the type of a Work Product?

Conceptual Model



Axiomatization:

-

WPC – Work Product Composition

Name: Work Product Composition (WPC)

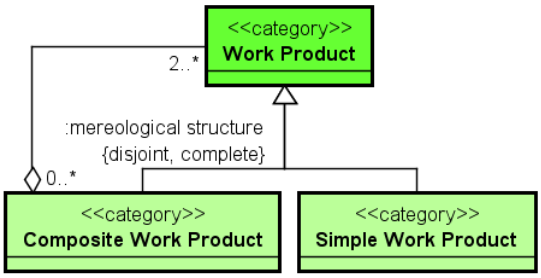
Intent: To represent the mereological structure of work products, distinguishing between composite and simple work products, and representing the decomposition of composite work products.

Rationale: *Work Products* may be decomposed in other *work products*. A *Composite Work Product* is composed of two or more *work products*. A *Simple Work Product* can be a part of a com*posite work product*, but cannot be decomposed.

Competency Questions:

- How Work Products are decomposed?

Conceptual Model



Axiomatization:

A1	∀w,c partOf(w,c) → (w != c)

- Axiom A1: A **Work Product** cannot be part of itself.

DocD – Document Depiction

Name: Document Depiction (DocD)

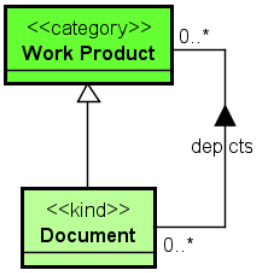
Intent: To represent the work products that a document depicts.

Rationale: A *Document* has relevant information about an endeavor. It can have independent information, or information about other *Work Products*. In the last case, we say that a *document* depicts *work products*.

Competency Questions:

- Which Work Products does a Document depict?

Conceptual Model



Axiomatization:

A1	∀a,b: depicts(a,b) → (a != b)

- Axiom A1: If a **Document** *a* depicts a **Work Product** *b*, then *a* is different from *b* (non-reflexibility).

WPP – Work Product Participation

Name: Work Product Participation (WPP)

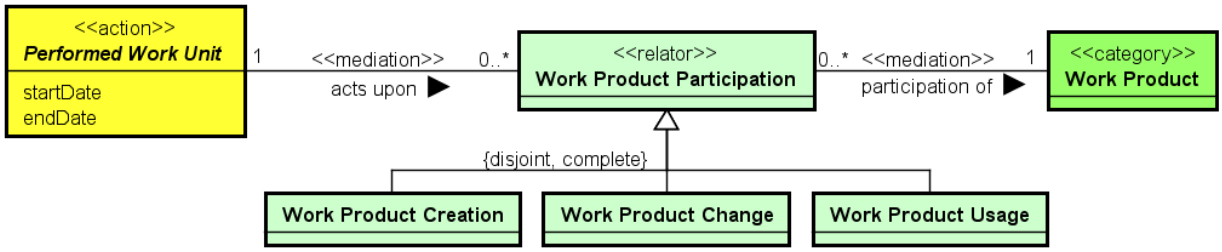
Intent: To represent the participation (creation, change or usage) of work products in performed work units.

Rationale: *Performed Work Units* handle *Work Products*, by *creating*, *changing* or *using* them.

Competency Questions:

- Which are the Work Products participating in a given Performed Work Unit?
- What are the types of the Work Product Participations?

Conceptual Model



Axiomatization:

- -

WPCrea – Work Product Creation

Name: Work Product Creation (WPCrea)

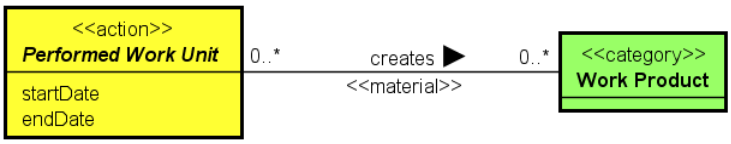
Intent: To represent the creation of work products by performed work units.

Rationale: *Performed Work Units* can create *Work Products* during their execution.

Competency Questions:

- What are the Work Products created by a Performed Work Unit?

Conceptual Model



Axiomatization:

-

WPUse – Work Product Usage

Name: Work Product Usage (WPUse)

Intent: To represent the usage of work products by performed work units.

Rationale: *Performed Work Units* can use *Work Products*.

Competency Questions:

- What are the Work Products used by a given Performed Work Unit?

Conceptual Model

Axiomatization:

-

WPChan – Work Product Change

Name: Work Product Change (WPChan)

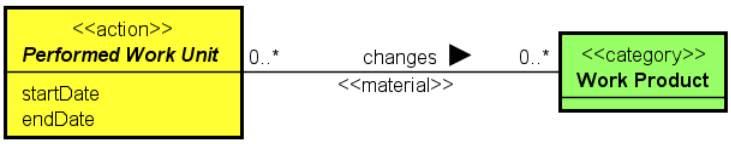
Intent: To represent the change of work products during the execution of performed work units.

Rationale: *Performed Work Units* can change *Work Products*.

Competency Questions:

- What are the Work Products changed by a given Performed Work Units?

Conceptual Model



Axiomatization:

-

References

[1] FALBO, R. A., BARCELLOS, M.P., NARDI, J.C., GUIZZARDI, G. Organizing ontology design patterns as ontology pattern languages. *Proc. 10th Extended Semantic Web Conference* (ESWC'13), Montpellier, France, 2013.

[2] RUY, F.B., FALBO, R.A., BARCELLOS, M.P., GUIZZARDI, G. An Ontological Analysis of the ISO/IEC 24744 Metamodel. *Proc. 8th International Conference on Formal Ontology in Information Systems (FOIS'14)*, Rio de Janeiro, Brazil, 2014.

[3] FALBO, R.A., RUY, F. B., GUIZZARDI, G., BARCELLOS, M. P., ALMEIDA, J. P. A. Towards an enterprise ontology pattern language. *Proc. 29th Annual ACM Symposium on Applied Computing*, Gyeongju, Korea, 2014, 323-330.