

Tecnologías para la Web Semántica

Ingeniería Ontológica III
Evaluación

Evaluación

- ▶ Se describe la evaluación como el análisis de una ontología con respecto a una referencia durante cada etapa de su desarrollo.
- ▶ Los términos verificación y validación se encuentran incluidos en la evaluación.
 - La verificación se relaciona con la determinación de si la ontología fue correctamente construida.
 - La validación permite determinar si la ontología captura correctamente el mundo real que se está modelando.

Evaluación

- ▶ El objetivo de este proceso es determinar si la ontología desarrollada cumple con los principios de diseño y con los requerimientos descritos en el DERO.
- ▶ Controlar que la ontología responde a las preguntas de competencia formuladas.

Evaluación

- ▶ **Verificación y Validación (V&V):** nombre dado a los procesos que permiten asegurar que la ontología desarrollada satisface su especificación y brinda la funcionalidad esperada por las personas que la solicitaron.
- ▶ *Sin embargo, no son lo mismo...*
 - **Validación:** ¿Estamos construyendo el producto *correcto*?
 - **Verificación:** ¿Estamos construyendo el producto *correctamente*?

Evaluación

- ▶ **Verificación:** mostrar que la ontología cumple con su funcionalidad y con los requerimientos no funcionales establecidos.
- ▶ **Validación:** mostrar que la ontología cumpla con las expectativas del cliente. Es más general.

Evaluación

- ▶ Realizar un control de la ontología para la detección de anomalías o malas prácticas en su diseño.
- ▶ Existen diversas propuestas que contienen buenas prácticas de diseño para tener en cuenta al momento de desarrollar una ontología.
- ▶ Existe un catálogo de errores comunes entre los que podemos encontrar:

Poveda-Villalón, M., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2012). Validating ontologies with oops!. In Knowledge Engineering and Knowledge Management (pp. 267-281). Springer Berlin Heidelberg.

Evaluación

- ▶ Crear sinónimos como clases.
 - Determinar clases equivalentes en vez de especificar sinonimia.
 - Dos términos sinónimos refieren a la misma clase, no a clases diferentes aunque sean equivalentes.
 - Dos términos sinónimos serían “Auto” y “Automóvil”.
- ▶ Utilizar la relación “es” en vez de utilizar elementos propios o primitivas de lenguajes semánticos como:
 - "rdfs:subClassOf" que expresa subclase
 - "rdf:type" que expresa membresía
 - "owl:sameAs" que expresa igualdad entre instancias.
- ▶ Creación de elementos no conectados.
 - Los elementos de la ontología están creados sin conexión con los elementos restantes de la ontología quedando miembros aislados.
 - Un ejemplo de este caso es la definición de la clase “IntegranteEquipo” y no contar en la ontología con la clase “Equipo”.

Evaluación

- ▶ Definición de relaciones inversas erróneas.
 - Definir relaciones como inversas cuando no necesariamente son inversas. Por ejemplo si algo se compra, “esComprado” no sería una relación inversa correcta.
- ▶ Mezclado de diferentes conceptos en la misma clase.
 - En este caso se crea una clase cuyo identificador hace referencia a dos o más conceptos.
 - Un ejemplo de este error es el identificador “ProductosyServicios”.
- ▶ Falta de anotaciones.
 - En este caso los términos de la ontología carecen de atributos.
 - Esta clase de propiedades facilitan la comprensión de la ontología y su usabilidad desde el punto de vista del usuario.

Evaluación

► Falta de disjunción.


- En este caso, a la ontología le falta la definición de axiomas de disjunción entre clases o entre propiedades que deberían ser definidas como disjuntas.
- Por ejemplo, se pueden crear las clases “par” e “impar” sin que sean disjuntas, pero esta representación no es correcta basado en la definición de este tipo de números.

Evaluación

- ▶ Herramienta OOPS! Ontology Pitfall Scanner colabora con la detección de los errores catalogados en ontologías (errores comunes).

<http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp>

① oops.linkeddata.es

 **OntOlogy Pitfall Scanner!**

OOPS! (OntOlogy Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.
To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI: Scanner by URI

Example: http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

Scanner by direct input:

If you just include the RDF code here, the following Pitfalls will not be checked:
P36. URI contains file extension, P37. Ontology not available, P40. Namespace hijacking

Scanner by RDF

☐ Uncheck this checkbox if you don't want us to keep a copy of your ontology.

[Go to advanced evaluation](#)

News!

Now you can **integrate OOPS!** pitfall detection **with your own developments and tools** simply by invoking the **OOPS! RESTful Web Service**.

Detecting common pitfalls in ontologies

Modelling ontologies has become one of the main topics of research within ontological engineering because of the difficulties it involves. Developers must tackle a wide range of difficulties and handicaps when modelling ontologies that can imply the appearance of anomalies or errors in ontologies. Therefore, it is important to evaluate the ontologies in order to detect those potential problems.

Want to help?

- [Suggest new pitfalls](#)
- [Provide feedback](#)

Documentation:

- [Pitfall catalogue](#)

Evaluación

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	1 case Minor 🟡
Results for P05: Defining wrong inverse relationships.	2 cases Critical 🚫
Results for P08: Missing annotations.	51 cases Minor 🟡
Results for P11: Missing domain or range in properties.	7 cases Important ⚠️
Results for P13: Missing inverse relationships.	7 cases Minor 🟡

This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".

- This pitfall appears in the following elements:
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#isComposedByReactive>
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#hasCoAssessed>
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#isReactiveOf>
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#hasSelfEvaluator>
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#hasCoEvaluator>
 - <http://www.owl-ontologies.com/Ontology1328898742.owl#hasSelfAssessed>
 - <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#hasRuleGroup>

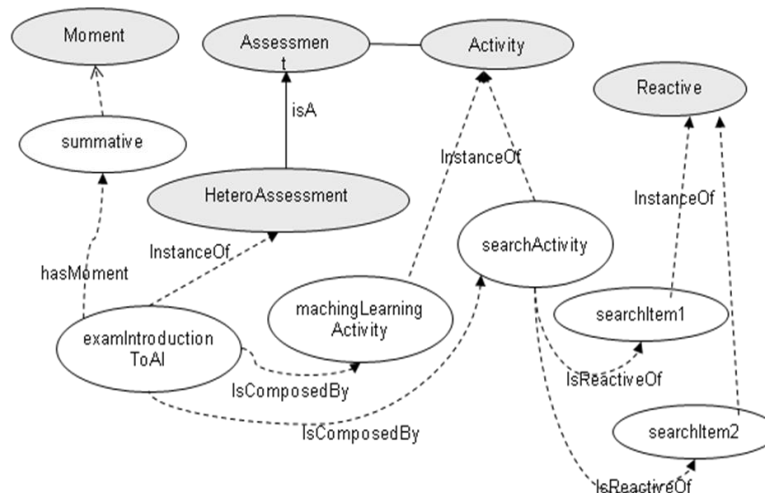
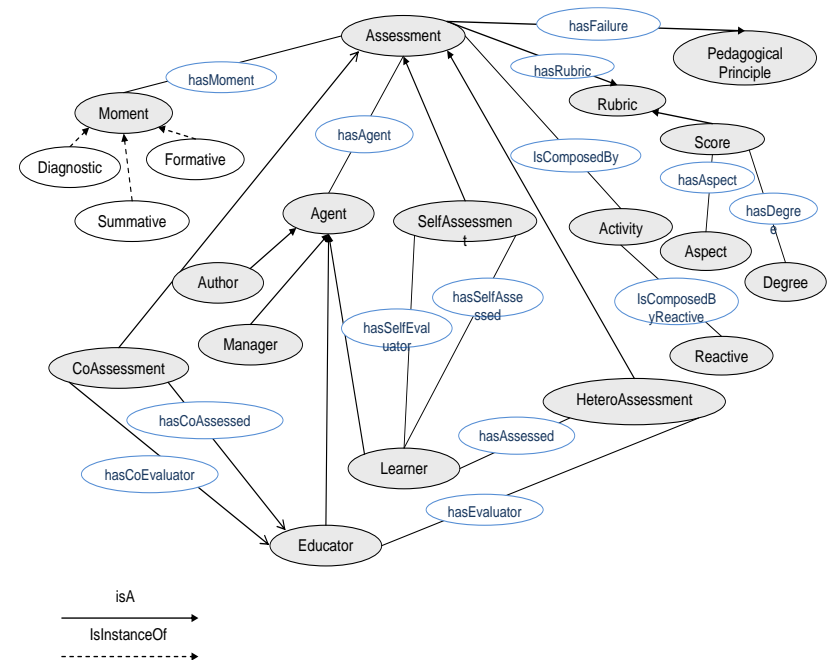
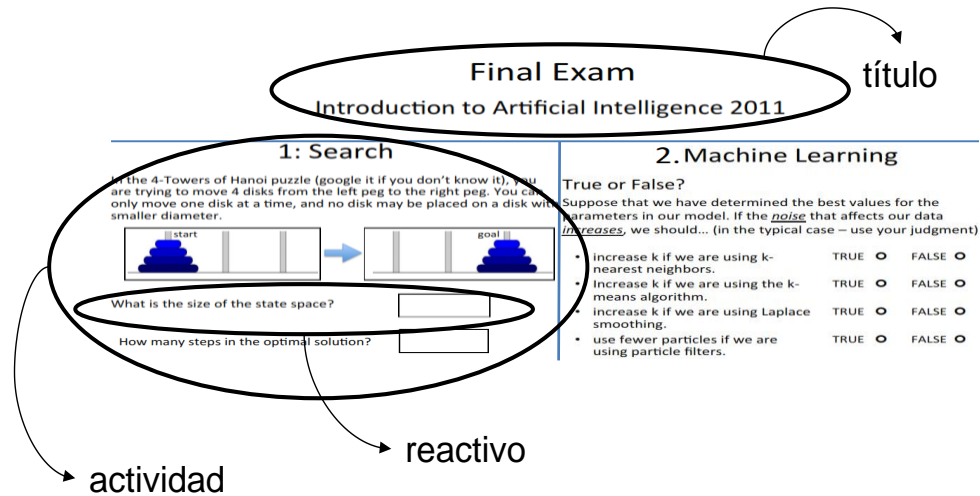
Evaluación

- ▶ El motor de inferencia Pellet se utiliza para verificar la consistencia formal de la ontología.
- ▶ Pellet se puede instalar como un plugin en Protégé ya que es totalmente compatible con el lenguaje OWL2.
- ▶ Las inconsistencias podrían presentarse relacionadas con la disposición de las clases (clases en la misma jerarquía y clases disjuntas), o pueden estar referidas a la relación entre las clases (rango y dominio), en el tipo de atributo o en las reglas de aplicación de la ontología.

Evaluación

- ▶ Con respecto a los requerimientos, la evaluación se realiza mediante el uso de las preguntas de competencia.
- ▶ Con este fin, la ontología debe instanciarse.
- ▶ Se consideran las preguntas de competencia realizadas en la especificación de requerimientos para ver si la ontología puede responderlas.
- ▶ Las respuestas obtenidas a las preguntas de competencia se deben someter a la consideración de expertos del dominio quienes deben determinar si son aceptables las repuestas

Ejemplo



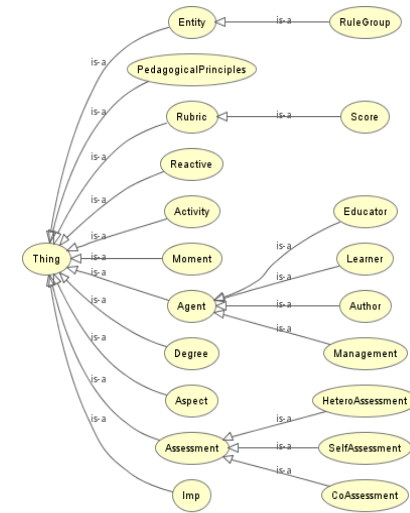
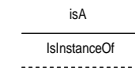
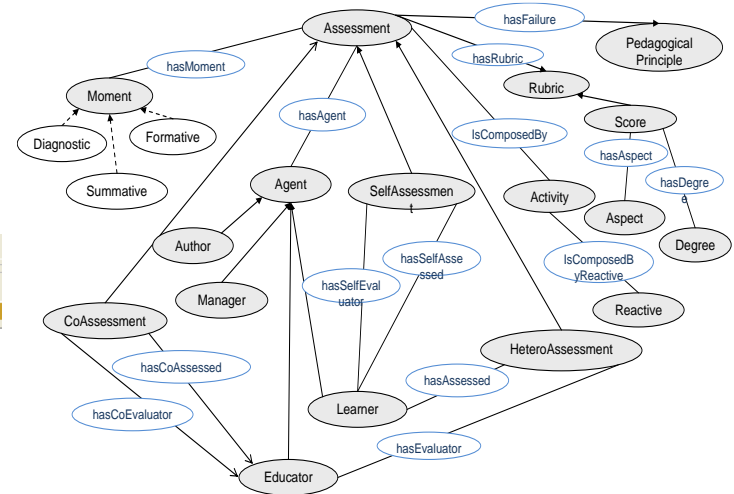
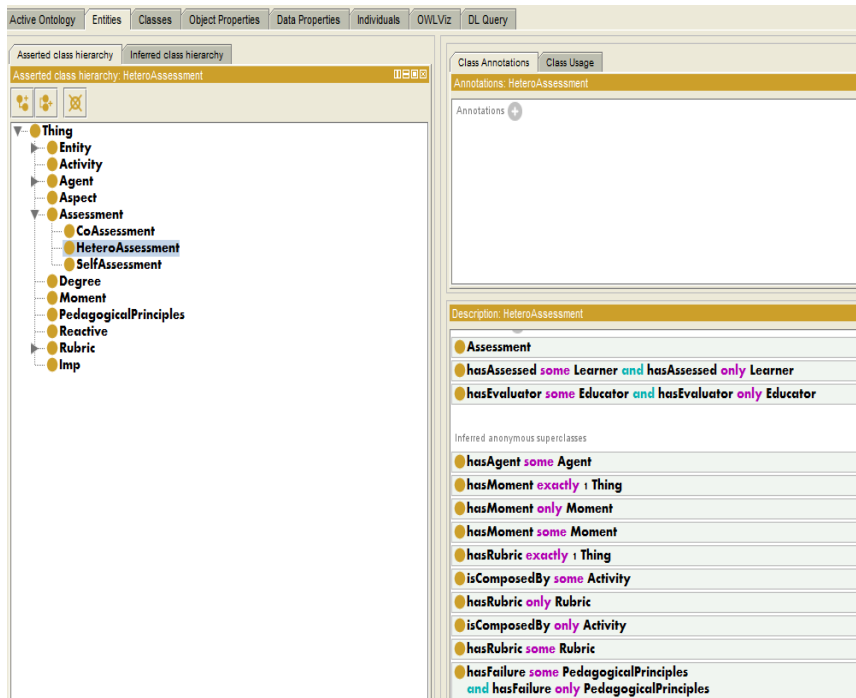
Ejemplo

- ▶ ¿Qué alumnos respondieron una evaluación?
- ▶ ¿Cuáles son las actividades de una heteroevaluación?

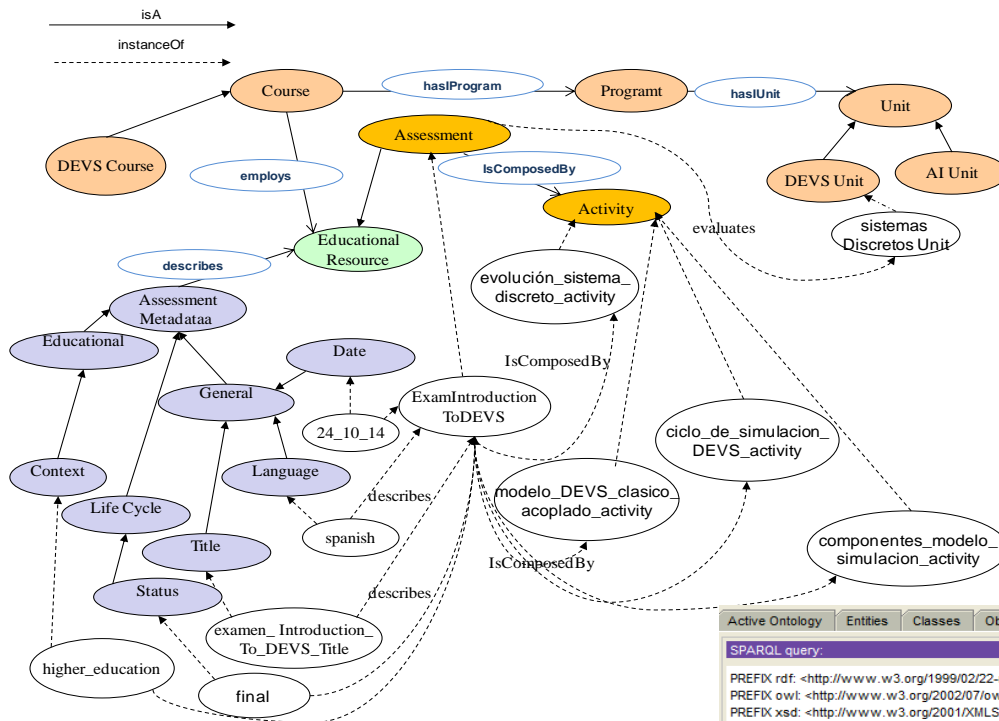
Active Ontology		Entities	Classes	Object Properties	Data Properties	Individuals	OWL Viz	DL Query	SPARQL Query
SPARQL query:									
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX asse: <http://www.owl-ontologies.com/Ontology1328898742.owl#> SELECT ?learner ?assessment WHERE { ?assessment asse:hasAssessed ?learner }									
learner					assessment				
Perez_Juan Gonzalez_Maria Perez_Juan Gonzalez_Maria					Assessment_1 Assessment_1 examIntroductionToAI examIntroductionToAI				

Active Ontology		Entities	Classes	Object Properties	Data Properties	Individuals	OWL Viz	DL Query	SPARQL Query
SPARQL query:									
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX asse: <http://www.owl-ontologies.com/Ontology1328898742.owl#> SELECT ?hetero ?activity WHERE { ?hetero rdf:type asse:HeteroAssessment. ?assessment asse:isComposedBy ?activity FILTER (?hetero = ?assessment) }									
hetero					activity				
examIntroductionToAI examIntroductionToAI					searchActivity machineLearningActivity				

Ejemplo



- ▶ CQ ¿Qué evaluación se tomo el día 24/10/14?



Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
OVLViz
DL Query
SPARQL Query

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX aonet: <http://www.owlim-ontologies.com/Ontology1328898742.owl#>
SELECT ?metadata
WHERE {
  ?assessment aonet:isDescribedBy ?metadata.
  FILTER regex(str(?assessment), 'examIntroductionToDEV5' )}

```

higher_education

final

spanish

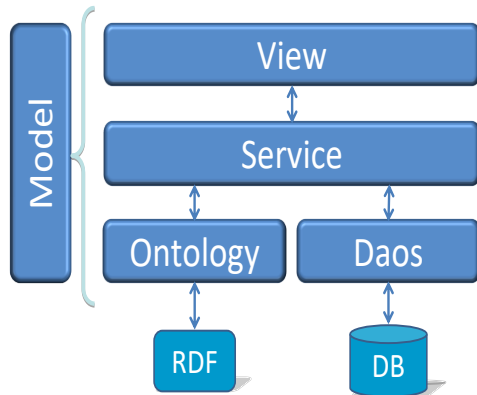
24_10_14

examen_introduction_to_DEV5_title

metadata

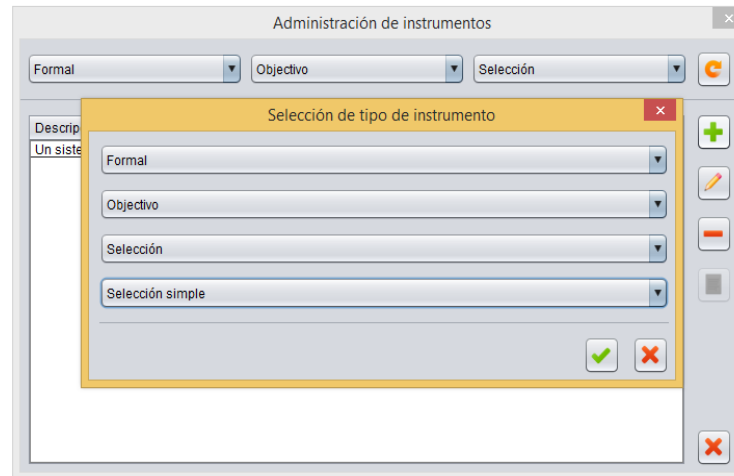
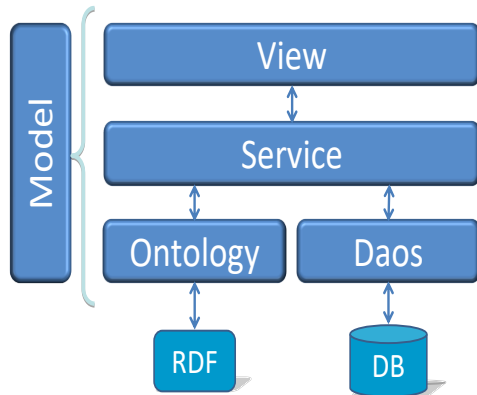
Evaluación

- ▶ Evaluación de una ontología mediante una aplicación.
- ▶ Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.



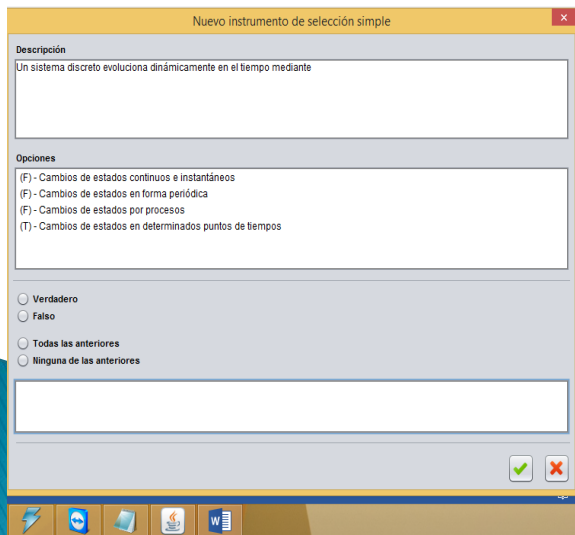
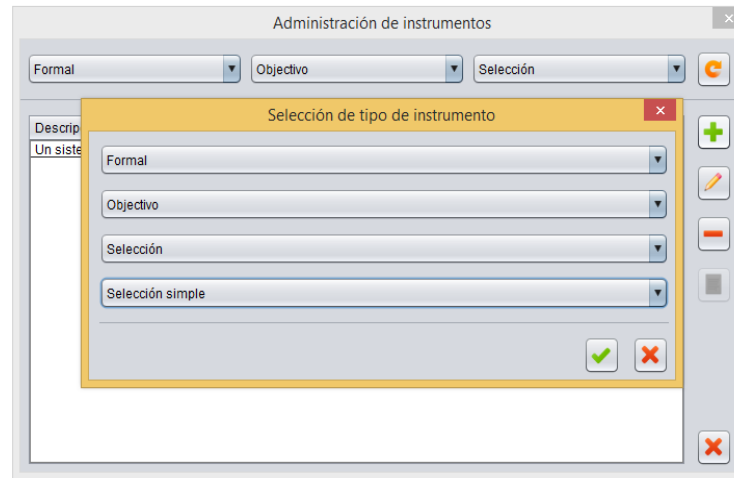
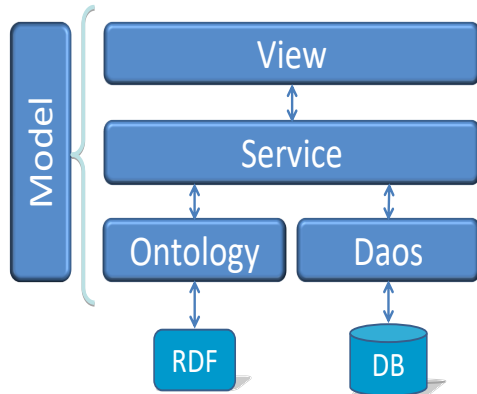
Evaluación

- ▶ Evaluación de una ontología mediante una aplicación.
- ▶ Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.



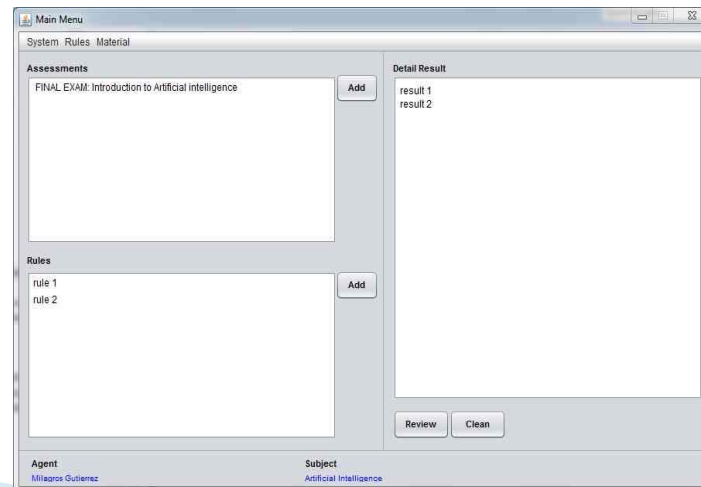
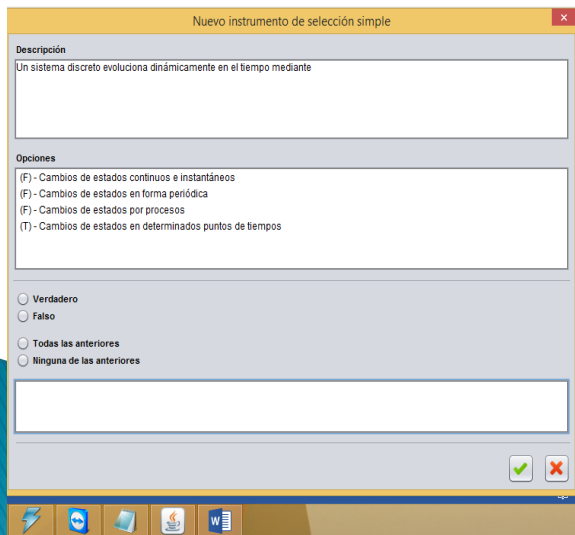
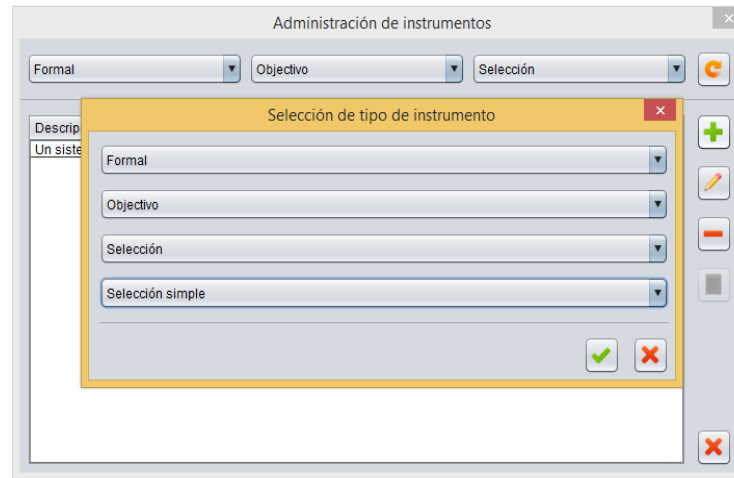
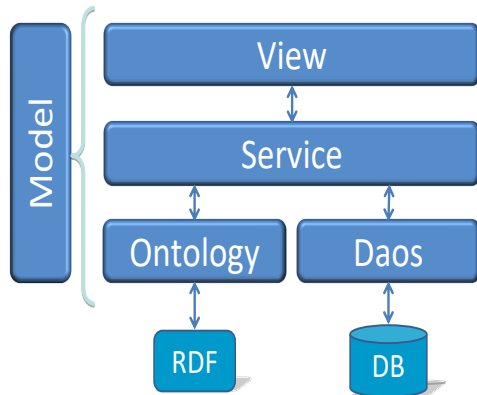
Evaluación

- ▶ Evaluación de una ontología mediante una aplicación.
- ▶ Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.



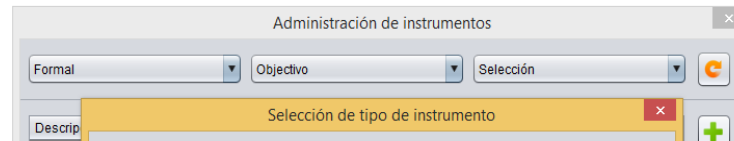
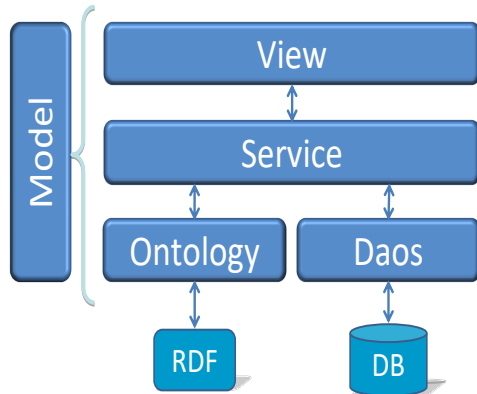
Evaluación

- ▶ Evaluación de una ontología mediante una aplicación.
- ▶ Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.



Evaluación

- ▶ Evaluación de una ontología mediante una aplicación.
- ▶ Durante el desarrollo de una ontología se realizan entrevistas con expertos. Es decir, las ontologías se evalúan desde su perspectiva de uso.



5. Un instrumento de tipo opciones múltiples no puede tener opciones como “todas las anteriores” o “ninguna de las anteriores”.

$$\mathfrak{S} | = (\forall x, y, z, w (MultipleChoices(x) \wedge hasOption(x, y) \wedge hasAttribute(y, z) \wedge value(z, w)) \Rightarrow (w \neq \text{"all of them"} \wedge w \neq \text{"none of them"}))$$
