

Universidad de Oviedo

# **Presentaciones Prueba 1 y Prueba 2**

**Plaza F036-570-DFA0340  
Área de Lenguajes y Sistemas Informáticos  
Departamento de Informática**

**Jose Emilio Labra Gayo**



# Primera prueba

## Historial académico, docente e investigador

### Proyecto docente

### Proyecto investigador

Jose Emilio Labra Gayo



Universidad de Oviedo

# Parte I

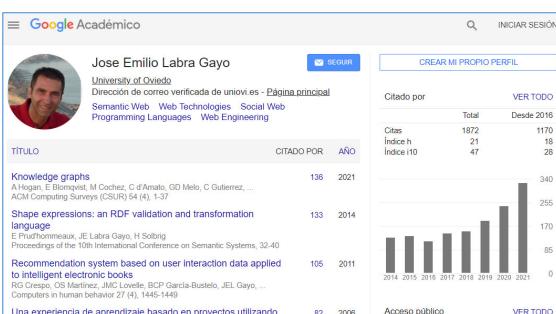
## Historial académico, docente e investigador

Jose Emilio Labra Gayo



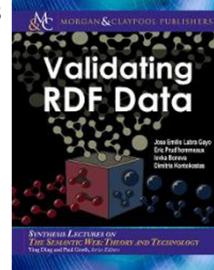
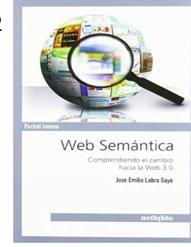
Universidad de Oviedo

Historial - datos generales	Proyecto docente	Proyecto investigador
<h1>Datos generales</h1>		
Formación académica		
<b>Doctor Ingeniero en Informática, Dpto. Informática, 2001 ( premio extraordinario)</b> <b>Ingeniero en Informática, ETS Ingenieros Industriales e Informáticos, 1994</b> <b>Diplomado en Informática, Sistemas, Esc. Ing. Téc. Informática, 1992</b>		
Actividad científico-profesional		
<b>Profesor Titular de Universidad, 2004</b> <b>Profesor Titular de Escuela Universitaria, 1999</b> <b>Profesor Asociado, 1992</b> <b>A. programador, 1990</b> <b>Becario, 1989-91</b>	<b>Investigador principal y fundador (2004)</b> <b>Grupo WESO (WEb Semantics Oviedo)</b> <b>Grupo reconocido en ANEP (2012,2018)</b> 	<b>Reconocimientos</b> <b>3 sexenios de investigación</b> <b>5 quinquenios de docencia</b>

Historial investigador	Proyecto docente	Proyecto investigador
<h1>Resumen Historial investigador</h1>		
<b>Según Scopus: h-index: 13</b> 95 documentos (629 citas de 528 documentos)		
<b>Según Google scholar: h-index: 21</b> 229 documentos, 1872 citas, i10-index: 47		
		

Histórico investigador	Proyecto docente	Proyecto investigador
<h2>Proyectos de investigación (conv. competitivas)</h2> <p><b>IP en 3 proyectos Plan Nacional</b></p> <ul style="list-style-type: none"> <li><a href="#">2021-23, ANGLIRU (Applying Knowledge Graphs to research data interoperability and ReUsability)</a></li> <li><a href="#">2017-20, VARSSHA (Validating RDF Streams using SHApes)</a></li> <li><a href="#">2012-14, ROCAS (ReasOning on the Cloud by applying Semantics)</a></li> </ul> <p><b>IP en colaboración con empresa, Compra pública innovadora</b></p> <ul style="list-style-type: none"> <li><a href="#">2020-21, Arquitectura Semántica e Infraestructura Ontológica, Proyecto Hércules, Univ. Murcia</a></li> </ul> <p><b>IP en 2 proyectos plan Avanza por parte Universidad de Oviedo</b></p> <ul style="list-style-type: none"> <li><a href="#">2012-14, TagFlow, sistema Avanzado de monitoreo y tagging de Flujos de datos multimedia, con Bmat</a></li> <li><a href="#">2010-11, Prototipo plataforma pan-europea servicios de información para PYMES en licitación pública internacional (10DERS INFORMATION SERVICES), con Gateway</a></li> </ul> <p><b>Miembro de equipo en proyecto Europeo</b></p> <ul style="list-style-type: none"> <li><a href="#">2010-12, Advancing the Multilingual Web</a></li> </ul>		

Histórico investigador	Proyecto docente	Proyecto investigador
<h2>Contratos investigación con empresas</h2> <p><b>IP en 23 contratos de investigación</b></p> <p><b>Instituciones y empresas con las que se colabora:</b></p> <ul style="list-style-type: none"> <li>University of Virginia, Sloan Foundation</li> <li>Micelio BVBA, Ministerio de Cultura de Luxemburgo</li> <li>Izertis S. A.</li> <li>Zapiens technologies S. L.</li> <li>Ayuntamiento de Avilés</li> <li>The Web Foundation</li> <li>SBC, International IFAD, Naciones Unidas (LandPortal)</li> <li>Biblioteca Nacional del Congreso de Chile</li> <li>BMat Licensing S. L.</li> <li>Treelogic</li> <li>Gateway S.C.S.</li> <li>Fundación CTIC y Oficina española W3C</li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> </ul>		

Historial investigador	Proyecto docente	Proyecto investigador
<h2>Publicaciones</h2> <p><b>Revistas</b></p> <p>35 revistas JCR</p> <ul style="list-style-type: none"> <li>16 Q1 (10 SCIE, 6 SSCIE)</li> <li>3 Q2</li> <li>11 Q3</li> <li>5 Q4</li> </ul> <p>4 revistas no JCR</p> <p><b>Libros</b></p> <ul style="list-style-type: none"> <li>3 libros como autor</li> <li>4 libros como editor</li> <li>7 capítulos de libros</li> </ul>	<p>2018</p> 	<p>2012</p> 

Historial investigador	Proyecto docente	Proyecto investigador
<h2>Congresos</h2> <p>40 publicaciones derivadas de congresos</p> <p>39 participaciones en congresos</p> <p>9 tutoriales impartidos en conferencias</p> <p>2 Best paper awards</p> <ul style="list-style-type: none"> <li>10th International conference on Semantics Systems, 2014</li> <li>16th Extended Semantic Web Conference (ESWC), 2019</li> </ul> <p>11 invitaciones a conferencias</p> <p>4 invitaciones como <i>Keynote Speaker</i></p> <ul style="list-style-type: none"> <li>HELDIG Digital Humanities Summit, 2019</li> <li>Iberoamerican Conf. on Knowledge Graphs and Semantic Web, 2019</li> <li>Workshop on semantic web and linked open data, 2013</li> <li>ESA-EUSC (European Space Agency-European Union Satellite Center) 2008</li> </ul>		

Historial investigador	Proyecto docente	Proyecto investigador
<h2>Hackathones y similares</h2> <div style="display: flex; justify-content: space-around;">    </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <p>Wikibase Summit, NYC, 2018      WikiCite, Berkeley, 2018      Wikimedia Hackathon, Praga, 2019</p> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;">   </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <p>Biohackathon, Fukuoka, Japón 2019      Virtual Biohackathon, Abril, Nov. 2020</p> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;">   </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <p>Biohackathon Europe, Barcelona, 2021      SWAT4HCLS Hackathon, Enero, 2020</p> </div>		

Historial docente	Proyecto docente	Proyecto investigador																																					
<h2>Docencia: 1er y 2º ciclo, grados</h2> <h3>Resumen asignaturas impartidas</h3> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070C0; color: white;">Asignatura</th> <th style="background-color: #0070C0; color: white;">Curso</th> <th style="background-color: #0070C0; color: white;">Titulación</th> <th style="background-color: #0070C0; color: white;">Desde</th> <th style="background-color: #0070C0; color: white;">Hasta</th> <th style="background-color: #0070C0; color: white;">Nº cursos</th> </tr> </thead> <tbody> <tr> <td rowspan="5" style="vertical-align: top; padding-right: 20px;">Coordinador/ responsablec</td> <td>Arquitectura del Software</td> <td>3º</td> <td>Grado Ingeniería Informática del Software</td> <td>2012-13</td> <td>Actualidad</td> <td>8</td> </tr> <tr> <td>Lógica</td> <td>1º</td> <td>Ingeniero técnico Informática Gestión/Sistemas</td> <td>1992-93</td> <td>2007-08</td> <td>15</td> </tr> <tr> <td>Programación lógica y funcional (Prog. declarativa)</td> <td>Opt.</td> <td>Ingeniero técnico Informática Gestión/Sistemas</td> <td>1994-95</td> <td>2007-08</td> <td>13</td> </tr> <tr> <td>Procesadores de lenguaje</td> <td>4º</td> <td>Ingeniero en Informática</td> <td>1996-97</td> <td>2004-05</td> <td>8</td> </tr> <tr> <td>Tecnologías Inform. Comunic. Historia del Arte</td> <td>Opt.</td> <td>Grado en Historia del Arte</td> <td>2011-12</td> <td>2019-20</td> <td>8</td> </tr> </tbody> </table> <p>Otras asignaturas: Computabilidad, Metodología de la programación, Fundamentos de Informática</p>			Asignatura	Curso	Titulación	Desde	Hasta	Nº cursos	Coordinador/ responsablec	Arquitectura del Software	3º	Grado Ingeniería Informática del Software	2012-13	Actualidad	8	Lógica	1º	Ingeniero técnico Informática Gestión/Sistemas	1992-93	2007-08	15	Programación lógica y funcional (Prog. declarativa)	Opt.	Ingeniero técnico Informática Gestión/Sistemas	1994-95	2007-08	13	Procesadores de lenguaje	4º	Ingeniero en Informática	1996-97	2004-05	8	Tecnologías Inform. Comunic. Historia del Arte	Opt.	Grado en Historia del Arte	2011-12	2019-20	8
Asignatura	Curso	Titulación	Desde	Hasta	Nº cursos																																		
Coordinador/ responsablec	Arquitectura del Software	3º	Grado Ingeniería Informática del Software	2012-13	Actualidad	8																																	
	Lógica	1º	Ingeniero técnico Informática Gestión/Sistemas	1992-93	2007-08	15																																	
	Programación lógica y funcional (Prog. declarativa)	Opt.	Ingeniero técnico Informática Gestión/Sistemas	1994-95	2007-08	13																																	
	Procesadores de lenguaje	4º	Ingeniero en Informática	1996-97	2004-05	8																																	
	Tecnologías Inform. Comunic. Historia del Arte	Opt.	Grado en Historia del Arte	2011-12	2019-20	8																																	

Historial docente		Proyecto docente		Proyecto investigador	
<b>Docencia: tercer ciclo y máster</b>					
Asignatura	Titulación	Desde	Hasta	Nº cursos	
Web Semántica	Máster en Ingeniería Web Universidad de Oviedo	2006-07	Actualidad	14	
Nuevos Avances en Web Semántica	Máster en Ingeniería Web Universidad de Oviedo	2006-07	Actualidad	14	
Sistemas y Aplicaciones Web	Magister en Tecnologías de la Información Universidad Técnica Federico Santa María, Chile	2005-06	Actualidad	15	
Web Semántica	Curso doctorado Dpto. Informática, Univ. Oviedo	2006-07	2010-11	4	
Lenguajes y estándares Web	Máster en Ingeniería Web Universidad de Oviedo	2006-07	2011-12	5	
Programación Orientada a Objetos	Máster en Ingeniería Web Universidad de Oviedo	2014-15	2018-19	4	
Web Semántica, estado del arte e innovaciones futuras	Doctorado en Ingeniería del Software Univ. Pontificia de Salamanca	2005-06	2007-08	2	

Historial docente		Proyecto docente		Proyecto investigador					
<b>Trabajos dirigidos</b>									
4 tesis doctorales dirigidas									
29 trabajos fin de máster, Máster Ingeniería Web									
16 trabajos fin de grado, Grado Ingeniería Informática del Software									
7 trabajos de investigación (doctorado)									
18 proyectos fin de carrera, Ingeniero en Informática									
59 proyectos fin de carrera, Ing. Téc. Informática Sistemas									
26 proyectos fin de carrera, Ing. Téc. Informática Gestión									

[Historial docente](#)[Proyecto docente](#)[Proyecto investigador](#)

## Otros méritos docentes

14 publicaciones en congresos de innovación docente

6 proyectos de innovación docente

9 publicaciones docentes

9 estancias en centros extranjeros

Acreditación para impartir docencia en inglés (2013)

La asignatura arquitectura del software se imparte en inglés/español

[Historial gestión](#)[Proyecto docente](#)[Proyecto investigador](#)

## Gestión

### Cargos unipersonales

**Director** de la Escuela de Ingeniería Informática 2004 a 2012 (8 años)

**Secretario** y Jefe de Estudios de dicha Escuela 1996 a 2000 (4 años)

**Director Cátedra** DXC Technologies, 2018 a actualidad

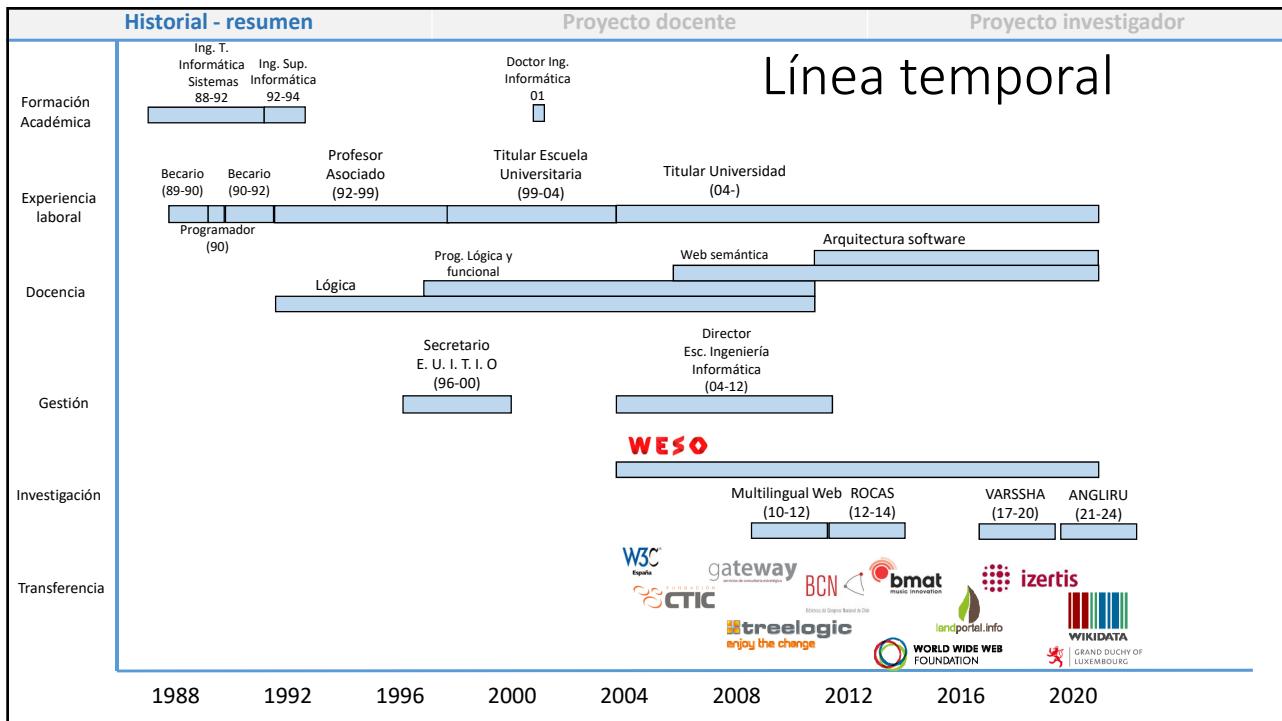
### Principales Órganos colegiados

2002-2008, 2017 - actualidad. Miembro Claustro, Univ. de Oviedo

2008-2012. Miembro Consejo de Gobierno, Univ. de Oviedo



Historial gestión	Proyecto docente	Proyecto investigador
<h2>Otras actividades de gestión/revisión</h2> <p>Evaluador proyectos europeos (4 ocasiones)</p> <p>21 revisiones de revistas según Publons</p> <p>Miembro de comité científico de 60 congresos</p> <p>Miembro comité organizador de 4 congresos</p> <p>Coordinador programas Erasmus e intercambio</p> <ul style="list-style-type: none"> <li>University of Malta</li> <li>Canisius College, Búfalo, Nueva York</li> <li>Kungliga Tekniska Högskolan, Suecia</li> <li>Universidade do Minho</li> </ul>		



# Parte II

## Proyecto docente

### "Arquitectura del Software"

Jose Emilio Labra Gayo



Universidad de Oviedo

[Historial](#)

[Proyecto docente](#)

[Proyecto investigador](#)

## Contenidos

### Entorno

- Universidad de Oviedo
- Departamento de Informática
- Escuela Ingeniería Informática
- Plan de estudios

### Concepto y referencias curriculares

- Qué es la arquitectura del software y evolución de la disciplina
- Referencias curriculares internacionales
- Arquitectura del software en otras universidades
- Relación con otras materias

### Propuesta docente

- Competencia y resultados de aprendizaje
- Unidades didácticas
- Organización: Clases de teoría, seminarios y prácticas de laboratorio
- Evaluación de la asignatura
- Justificación y posibles variaciones

Historial      Proyecto docente - entorno      Proyecto investigador

## Universidad de Oviedo

Fundada por Fernando Valdés Salas  
Inaugurada el 21 de Septiembre de 1608  
Décima Universidad española más antigua  
Única Universidad española creada en s. XVII

Primeros estudios:  
Artes, Cánones, Leyes, Teología (1608)



Historial      Proyecto docente - entorno      Proyecto investigador

## Universidad de Oviedo - actualidad

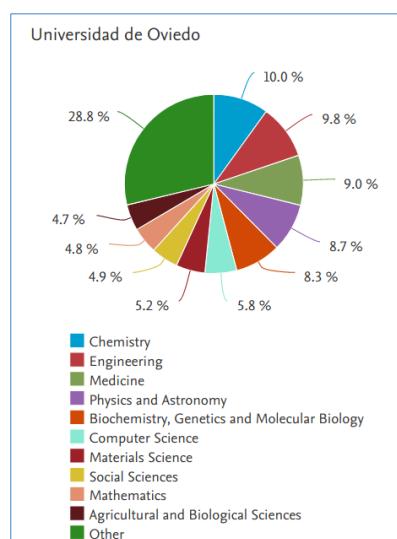
7 campus en 3 ciudades (Oviedo, Gijón, Mieres)  
38 Departamentos, 11 Facultades, 6 Escuelas, 3 Centros adscritos y 9 Institutos Universitarios  
22.000 estudiantes  
2.000 investigadores e investigadores docentes  
1.000 trabajadores de administración y servicios  
1500 estudiantes internacionales



**Universidad de Oviedo: Oferta educativa**

<b>61 grados</b> <ul style="list-style-type: none"> <li>9 Artes y humanidades</li> <li>7 Ciencias</li> <li>8 Ciencias de la Salud</li> <li>17 Ciencias sociales y Jurídicas</li> <li>20 Ingenierías, entre ellos: <i>Grado en Ingeniería Informática del Software</i></li> </ul>	<b>62 títulos de máster</b> <ul style="list-style-type: none"> <li>6 Erasmus Mundus</li> <li>8 internacionales</li> <li>7 interuniversitarios</li> <li>41 académico-profesionalizantes</li> <li>3 Artes y humanidades</li> <li>6 Ciencias</li> <li>7 Ciencias de la salud</li> <li>10 Ciencias sociales y jurídicas</li> <li>15 Ingeniería y arquitectura, entre ellos: Máster Universitario en Ingeniería informática Máster Universitario en Ingeniería Web</li> </ul>	<b>26 programas de doctorado</b> <ul style="list-style-type: none"> <li>3 artes y humanidades</li> <li>8 ciencias</li> <li>3 ciencias de la salud</li> <li>5 ciencias sociales y jurídicas</li> <li>9 rama ingeniería y arquitectura</li> <li>Entre ellos: <i>Programa doctorado en Informática</i></li> </ul>
--	--	--

**Universidad de Oviedo: Investigación**

<b>Campus de Excelencia Internacional: 2009</b> <b>166 grupos de investigación</b> <ul style="list-style-type: none"> <li>40 en Ciencias</li> <li>24 en Ciencias de la Salud</li> <li>38 en Ciencias Sociales y Jurídicas</li> <li>28 en Artes y Humanidades</li> <li>36 en la rama de Ingeniería y Arquitectura</li> </ul>	 <p style="text-align: center;">Universidad de Oviedo</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th>Área</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr><td>Chemistry</td><td>10.0 %</td></tr> <tr><td>Engineering</td><td>9.8 %</td></tr> <tr><td>Medicine</td><td>9.0 %</td></tr> <tr><td>Physics and Astronomy</td><td>8.7 %</td></tr> <tr><td>Biochemistry, Genetics and Molecular Biology</td><td>8.3 %</td></tr> <tr><td>Computer Science</td><td>5.8 %</td></tr> <tr><td>Materials Science</td><td>5.2 %</td></tr> <tr><td>Social Sciences</td><td>4.9 %</td></tr> <tr><td>Mathematics</td><td>4.8 %</td></tr> <tr><td>Agricultural and Biological Sciences</td><td>4.7 %</td></tr> <tr><td>Other</td><td>28.8 %</td></tr> </tbody> </table> <p style="text-align: center;">Producción científica por áreas. Fuente: Scopus</p>	Área	Porcentaje	Chemistry	10.0 %	Engineering	9.8 %	Medicine	9.0 %	Physics and Astronomy	8.7 %	Biochemistry, Genetics and Molecular Biology	8.3 %	Computer Science	5.8 %	Materials Science	5.2 %	Social Sciences	4.9 %	Mathematics	4.8 %	Agricultural and Biological Sciences	4.7 %	Other	28.8 %
Área	Porcentaje																								
Chemistry	10.0 %																								
Engineering	9.8 %																								
Medicine	9.0 %																								
Physics and Astronomy	8.7 %																								
Biochemistry, Genetics and Molecular Biology	8.3 %																								
Computer Science	5.8 %																								
Materials Science	5.2 %																								
Social Sciences	4.9 %																								
Mathematics	4.8 %																								
Agricultural and Biological Sciences	4.7 %																								
Other	28.8 %																								

Historial	Proyecto docente - entorno	Proyecto investigador
-----------	----------------------------	-----------------------

**Departamento de Informática**

Creado en 1996  
**146 profesores en 4 áreas de conocimiento**  
 Lenguajes y Sistemas Informáticos (69)  
 Ciencias de la Computación e Inteligencia Artificial (50)  
 Arquitectura y Tecnología de Computadores (14)  
 Ingeniería Telemática (13)

**12 CU, 52 TU, 6 CEU, 5 TEU**  
**3 PAS**

<https://www.di.uniovi.es/>



Historial	Proyecto docente - entorno	Proyecto investigador
-----------	----------------------------	-----------------------

**Escuela Ingeniería Informática**

Etapa	Director	Hitos
1982-1987	D. Florentino Braña Valdés	Inauguración (1982)
1987-1996	D. Andra Huerga Alonso	De diplomatura a ingeniería técnica (1992) Nuevo nombre: "Escuela Universitaria Ingeniería Técnica Informática" (1992)
1996-2004	D. Juan Manuel Cueva Lovelle	Nuevo plan de estudios (2002) Traslado edificio Valdés Salas (2004)
2004-2012	D. Jose Emilio Labra Gayo	Máster Ingeniería Web (2007) Plan estudios Bolonia (2010) Nuevo nombre: "Escuela Ingeniería Informática" (2010)
2012-2020	D. Benjamín López Pérez	Sello calidad Euro-Inf EQANIE (2016)
2020-	D. Fernando Álvarez García	



Historial	Proyecto docente - entorno	Proyecto investigador												
<h1>Grado Ingeniería Informática del Software</h1> <p>Adaptado EEEES          Aprobado en 2010          Comienza docencia en 2010-11          Grado bilingüe          Duración: 4 años          Acreditación ANECA renovada en 2016          Obtención sello calidad Euro-Inf EQANIE</p>														
		<table border="1"> <tr><td>Total créditos ECTS</td><td>240</td></tr> <tr><td>Formación básica</td><td>60</td></tr> <tr><td>Obligatorias</td><td>144</td></tr> <tr><td>Optativas</td><td>18</td></tr> <tr><td>Prácticas externas obligatorias</td><td>6</td></tr> <tr><td>Trabajo fin de grado</td><td>12</td></tr> </table>	Total créditos ECTS	240	Formación básica	60	Obligatorias	144	Optativas	18	Prácticas externas obligatorias	6	Trabajo fin de grado	12
Total créditos ECTS	240													
Formación básica	60													
Obligatorias	144													
Optativas	18													
Prácticas externas obligatorias	6													
Trabajo fin de grado	12													
														

Historial	Proyecto docente - entorno	Proyecto investigador															
<h2>Plan de estudios</h2> <p>4 cursos (2 semestres)          9 asignaturas optativas</p>																	
	<table border="1"> <thead> <tr> <th></th> <th>Semestre 1</th> <th>Semestre 2</th> </tr> </thead> <tbody> <tr><td>1º</td><td>Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación</td><td>Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación</td></tr> <tr><td>2º</td><td>Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad</td><td>Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia</td></tr> <tr><td>3º</td><td>Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I Sistemas Inteligentes</td><td>Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación</td></tr> <tr><td>4º</td><td>Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III</td><td>Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado</td></tr> </tbody> </table>		Semestre 1	Semestre 2	1º	Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación	Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación	2º	Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad	Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia	3º	Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I Sistemas Inteligentes	Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación	4º	Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III	Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado	
	Semestre 1	Semestre 2															
1º	Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación	Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación															
2º	Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad	Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia															
3º	Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I Sistemas Inteligentes	Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación															
4º	Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III	Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado															

[Historial](#)[Proyecto docente - concepto y referencias curriculares](#)[Proyecto investigador](#)

## Concepto y referencias curriculares

Definición de arquitectura del software

Evolución de la disciplina

Referencias curriculares

Arquitectura del software en otras universidades

Relación con otras asignaturas del plan de estudios

[Historial](#)[Proyecto docente - concepto y referencias curriculares](#)[Proyecto investigador](#)

## ¿Qué es la arquitectura del Software?

Definición ISO/IEC/IEEE 42010:2011

*Arquitectura: conceptos o propiedades fundamentales de un sistema en su entorno, representado por sus elementos, las relaciones entre ellos, y los principios de su diseño y evolución.*

*"Architecting": proceso de concebir, definir, expresar, documentar, comunicar, certificar una implementación apropiada, mantener y mejorar una arquitectura a través del ciclo de vida de un sistema*

Muchas otras definiciones como:

*"Toda arquitectura es diseño pero no todo diseño es arquitectura.*

*Arquitectura representa las decisiones de diseño significativas que dan forma al sistema, donde significativo se mide por el coste de cambio" (G. Booch)*

Historial | Proyecto docente - concepto y referencias curriculares | Proyecto investigador

## Evolución de la arquitectura del software

Arquitectura del software = metáfora de "arquitectura"

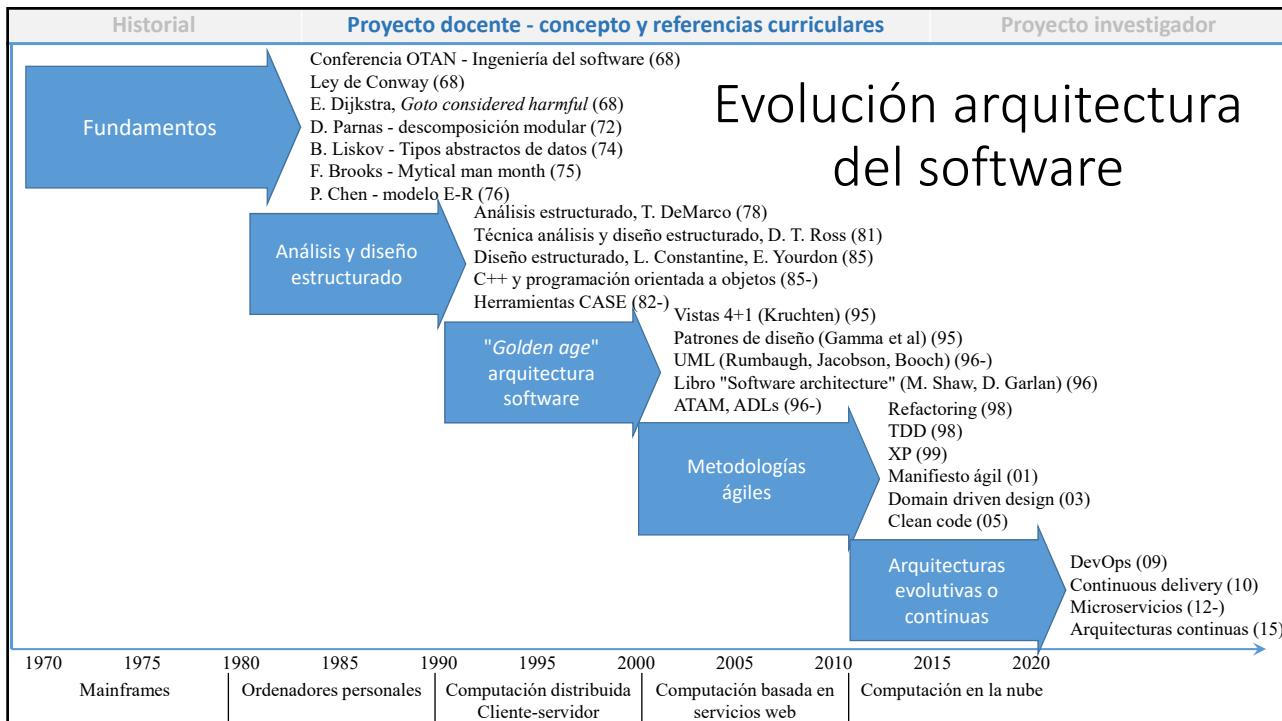
Etimológicamente: Arquitectura  $\rightsquigarrow$  ἀρχιτέκτων, "creador jefe"

Vitruvio, "De arquitectura", 3 pilares de una arquitectura

- Utilitas (utilidad)
- Firmitas (durabilidad)
- Venustas (elegancia)



Architectura (wikipedia): *Proceso y producto de planificar, diseñar y construir edificios u otras estructuras*



[Historial](#) | [Proyecto docente - concepto y referencias curriculares](#) | [Proyecto investigador](#)

# ¿Futuro de la arquitectura del software?

Actualmente:

- Microservicios y creciente interés en aspectos de arquitectura del software

¿Perspectivas de futuro?

- Más allá de la computación en la nube: ¿Cloud/edge/fog computing, Computación ubicua?

Otros atributos de calidad que cobrarán importancia:

- Privacidad, aspectos éticos y morales, sostenibilidad, ...

Este diagrama muestra la evolución de la arquitectura del software a lo largo del tiempo, dividida en períodos históricos y sus respectivos atributos de calidad.

Atributo de calidad	Periodo
Complejidad Rendimiento	1970 - 1975 (Mainframes)
Usabilidad	1980 - 1985 (Ordenadores personales)
Disponibilidad	1990 - 1995 (Computación distribuida Cliente-servidor)
Estabilidad Escalabilidad	2000 - 2005 (Computación basada en servicios web)
Seguridad privacidad	2010 - 2015 (Computación en la nube)
Aspectos éticos y morales Sostenibilidad	2020 - Actualidad (¿Edge/Fog comp. Computación ubicua?)

[Historial](#) | [Proyecto docente - concepto y referencias curriculares](#) | [Proyecto investigador](#)

## Referencias curriculares

- ACM-IEEE Computing Curricula (SEEK 2014)
- SWBOK v3 (2014)
- Otros
  - IFIP Informatics Curricula 2000
  - IASA Global ITABoK v3
  - iSAQB
- Otras asignaturas
  - Internacionales
  - Nacionales

Logos de las organizaciones profesionales ACM (Association for Computing Machinery) y IEEE (Institute of Electrical and Electronics Engineers).

Historial | Proyecto docente - concepto y referencias curriculares | Proyecto investigador

# SE2014 - ACM-IEEE Computing Curricula




## Software Engineering Computing Curricula (SE2014)

Unidad "Architectural design" (DES.ar) dentro del área "Software design"

DES.ar	Architectural design		
DES.ar.1	Marcos, estilos y patrones arquitectónicos	a	E
DES.ar.2	Compromisos arquitectónicos entre varios atributos	a	E
DES.ar.3	Temas sobre ingeniería de sistemas y hardware en arquitectura del software	k	E
DES.ar.4	Trazabilidad de requisitos en arquitectura	k	E
DES.ar.5	Arquitecturas orientadas a servicios	k	E
DES.ar.6	Arquitecturas para sistemas en red, móviles y empotrados	K	E
DES.ar.7	Relación entre arquitectura de producto, estructura de la organización de desarrollo y mercado	k	E

Clave: a (aplicación), k (conocimiento), E (esencial)

Historial | Proyecto docente - concepto y referencias curriculares | Proyecto investigador

# SWBoK v3 (2014)

IEEE Computer Society, 2014  
15 áreas de conocimiento

Ingeniería de requisitos software	Fundamentos diseño de software
Diseño de software	Temas clave en diseño de software
Construcción del software	Estructura y arquitectura del software
Pruebas del software	Diseño de interfaces de usuario
Mantenimiento de software	Análisis y evaluación de la calidad de diseño de software
Gestión de la configuración	Notaciones de diseño de software
Gestión de ingeniería de software	Métodos y estrategias de diseño de software
Proceso de ingeniería de software	Herramientas de diseño de software
Herramientas y métodos de ingeniería del software	Estructura y puntos de vista arquitectónicos
Calidad del software	Estilos arquitectónicos
Práctica profesional de la ingeniería del software	Patrones de diseño
Economía del software	Familias de programas y frameworks
Fundamentos de computación	
Fundamentos matemáticos	
Fundamentos de ingeniería	



Histórico | Proyecto docente - concepto y referencias curriculares | Proyecto investigador

## Otras referencias curriculares

IFIP Informatics Curricula (ICF 2001)  
MAP-03 (arquitectura de sistemas software)  
MAP-13 (desarrollo de software)

IASA Global (Asociación Internacional de arquitectos IT)  
ITABoK 3.0  
5 pilares: Estrategia de negocios, dinámicas humanas, diseño, Entorno TI, atributos de calidad  
4 especialidades: arquitectura de negocios, información, infraestructura, software

iSAQB (International Software Architecture Qualification Board)  
Certificados CPSA-Foundational, advanced  
Libro "Software architecture fundamentals", 2019



Histórico | Proyecto docente - concepto y referencias curriculares | Proyecto investigador

## ¿Cómo se imparte Arquitectura del software?

Dentro de asignatura *Ingeniería del Software*  
Sobre todo en titulaciones de ingeniería informática

Unida a otras asignaturas  
Ejemplo: "Arquitectura y Diseño software"

Asignatura independiente "Arquitectura del software"  
Suele estar complementada con una asignatura sobre diseño de software  
Modalidad Obligatoria/Optativa

A nivel de Máster

[Historial](#)[Proyecto docente - concepto y referencias curriculares](#)[Proyecto investigador](#)

## ASW en otras universidades (internacionales)

### Dentro de Ingeniería del Software

Ejemplos:

Rochester Institute of Technology: "Software architecture" (optativa)

### Unida a otras asignaturas

Ejemplos

CMU y Georgia Tech: *Software Architecture and Design*

### Asignatura específica "Arquitectura del Software"

Ejemplos:

Univ. de Alberta (Coursera): "Software architecture"

[Historial](#)[Proyecto docente - concepto y referencias curriculares](#)[Proyecto investigador](#)

## ASW en otras universidades (nacionales)

### Dentro de Ingeniería del Software

Ejemplos:

UCLM: "Diseño de Software", "Ingeniería del Software I", "Ingeniería del Software II"

### Unida a otras asignaturas

Ejemplos

UPM: "Arquitectura y diseño software"

U. Sevilla: "Arquitectura e integración de sistemas software"

### Asignatura específica "Arquitectura del Software"

Ejemplos:

UPC. "Arquitectura del software"

Historial	Proyecto docente - concepto y referencias curriculares	Proyecto investigador																							
<b>Relación con otras materias</b>																									
Referencia: Grado en Ingeniería del Software																									
<b>Verde:</b> Principales asignaturas que influyen ASW <b>Rojo:</b> asignaturas impartidas a la vez <b>Marrón:</b> asignaturas que son influídas por ASW	<table border="1"> <thead> <tr> <th></th> <th>Semestre 1</th> <th>Semestre 2</th> </tr> </thead> <tbody> <tr> <td>1º</td> <td>Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación</td> <td>6 6 6 6 6</td> <td>Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación</td> <td>6 6 6 6 6</td> </tr> <tr> <td>2º</td> <td>Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad</td> <td>6 6 6 6 6</td> <td>Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia</td> <td>6 6 6 6 6</td> </tr> <tr> <td>3º</td> <td>Repositorys de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I</td> <td>6 6 6 6 6</td> <td>Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I</td> <td>6 6 6 6 6</td> </tr> <tr> <td>4º</td> <td>Sistemas Inteligentes Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III</td> <td>6 6 6 6 6</td> <td>Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado</td> <td>6 6 6 6 6 6 6 6 6 12</td> </tr> </tbody> </table>		Semestre 1	Semestre 2	1º	Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación	6 6 6 6 6	Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación	6 6 6 6 6	2º	Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad	6 6 6 6 6	Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia	6 6 6 6 6	3º	Repositorys de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I	6 6 6 6 6	Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I	6 6 6 6 6	4º	Sistemas Inteligentes Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III	6 6 6 6 6	Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado	6 6 6 6 6 6 6 6 6 12	
	Semestre 1	Semestre 2																							
1º	Álgebra lineal Cálculo Empresa Fund. de informática Introd. a la programación	6 6 6 6 6	Ondas y electromagnetismo Estadística Fund. Computadores y redes Autómatas y matem. Discretas Metod. De la programación	6 6 6 6 6																					
2º	Tecn. Electrónica computadores Arquitectura de computadores Estructuras de datos Tecn. y parad. de programación Computabilidad	6 6 6 6 6	Sistemas operativos Comunicación persona-máquina Bases de datos Computación numérica Algoritmia	6 6 6 6 6																					
3º	Repositorys de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I	6 6 6 6 6	Repositorios de información Software y estándares para la web Ingeniería del proceso software Diseño del software Optativa I	6 6 6 6 6																					
4º	Sistemas Inteligentes Ingeniería de requisitos Calidad, Valid. y verif. Software Optativa II Optativa III	6 6 6 6 6	Sistemas distribuidos e internet Admon. Sistemas y Redes Seguridad Sistemas Informáticos Arquitectura del Software Diseño Lenguajes Programación Dirección y Plan. Proyectos Infor. Asp. Sociales legales, éticos y profesionales Informática Prácticas externas Trabajo fin de grado	6 6 6 6 6 6 6 6 6 12																					



Historial | Proyecto docente - propuesta docente | Proyecto investigador

# Propuesta docente "Arquitectura del software"

Curso: 3º, semestre 2  
Carácter obligatorio  
Créditos: 6  
Tipo: bilingüe (español/inglés)



Historial | Proyecto docente - propuesta docente | Proyecto investigador

## Competencias

Competencias Generales Instrumentales: Habilidades Metodológicas

**CG-1** Competencia para el diseño de soluciones a problemas complejos humanos

Competencias Específicas – Común a la rama de Informática

**Com.1** Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

**Com.8** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

**Com.11** Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.

Competencias Específicas – Tecnología Específica Ingeniería del Software

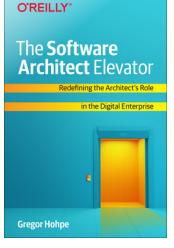
**ISW.1** Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.

**ISW.3** Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles

**ISW.4** Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Resultados de aprendizaje</h2>		
<b>RA.IS-1.</b>	Realizar Proyectos de Ingeniería del Software complejos que den solución a problemas reales y solucionarlos mediante técnicas y tecnologías relacionadas con los procesos de fabricación de software incluyendo <i>frameworks</i> , patrones arquitectónicos, patrones de diseño y de integración persiguiendo el desarrollo de software de calidad [Com1], [Com2], [Com.8], [Com10], [Com13], [ISw.1], [ISw.4], [ISw.6], [Bas5] , [CG1], [CG3],[CG4],[CG5],[CG10], [CG22]	
<b>RA.IS-3.</b>	Aplicar distintas técnicas de construcción de software en el diseño de bajo nivel [Com.8], [ISw.1], [ISw.4], [CG1]	
<b>RA.IS-4.</b>	Desarrollar diseños y programación orientados a objetos con un elevado nivel de competencia [Com.8], [ISw.4], [CG1], [CG4]	
<b>RA.IS-5.</b>	Evolucionar y refactorizar diseños existentes ante cambios en los requisitos [ISw.1], [ISw.6]	
<b>RA.IS-6.</b>	Determinar el grado de mantenibilidad, fiabilidad y eficiencia de diseños software [Com.8], [ISw.1], [ISw.4], [CG1]	
<b>RA.IS-7</b>	Diseñar e implementar software aplicando diferentes tecnologías middleware [Com11], [ISw3]	
<b>RA.IS-9</b>	Diseñar y llevar a cabo controles e inspecciones eficientes y efectivas de validación, verificación y calidad, planes de pruebas [Com1], [Com16], [ISw1], [ISw4], [CG2], [CG4], [CG11], [CG28].	
<b>RA.IS-10</b>	Analizar estadísticamente la densidad de defectos y probabilidad de fallos [Com16], [ISw4], [ISw5], [CG25]	
<b>RA.IS-11</b>	Evaluar la calidad de un proceso de software desde el punto de vista de la calidad del producto [Com1], [Com18], [ISw1], [CG2], [CG11], [CG28].	

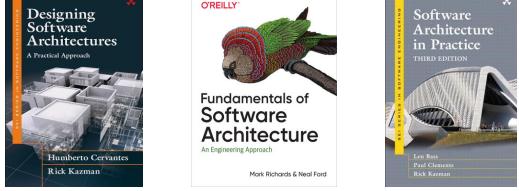
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Contenidos</h2>		
<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <ul style="list-style-type: none"> <li>Unidad 1. Definiciones de Arquitectura de Software</li> <li>Unidad 2. Comunicando la Arquitectura del Software. Documentación</li> <li>Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i></li> <li>Unidad 4. Atributos de calidad</li> <li>Unidad 5. Creación y validación de Arquitecturas</li> <li>Unidad 6. Técnicas de arquitectura del software: Tácticas, estilos y patrones arquitectónicos</li> </ul> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <ul style="list-style-type: none"> <li>Unidad 7. Construcción, Gestión de configuraciones</li> <li>Unidad 8. Modularidad</li> <li>Unidad 9. Comportamiento: monolito</li> <li>Unidad 10. Comportamiento: sistemas distribuidos y Big data</li> <li>Unidad 11. Disposición</li> <li>Unidad 12. Arquitectura del software y Entorno empresarial</li> </ul>		

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 1: Definiciones</h2> <p>¿Qué es la arquitectura del software?</p> <p>Proceso de creación de arquitectura</p> <ul style="list-style-type: none"> <li>Objetivos de diseño</li> <li>Requisitos funcionales</li> <li>Atributos de calidad</li> <li>Restricciones</li> <li>Preocupaciones</li> </ul> <p>Tipos de sistemas</p> <p><i>Greenfield, brownfield, ...</i></p>	<p>Tema 1. Conceptos de Arquitectura de Software</p> <p><b>Unidad 1. Definiciones</b></p> <p>Unidad 2. Comunicando la Arquitectura del Software</p> <p>Unidad 3. Papel del Arquitecto del Software y Stakeholders</p> <p>Unidad 4. Atributos de calidad</p> <p>Unidad 5. Creación y validación de Arquitecturas</p> <p>Unidad 6. Técnicas de arquitectura del software</p> <p>Tema 2. Taxonomías de Arquitectura del Software</p> <p>Unidad 7. Construcción, Gestión de configuraciones</p> <p>Unidad 8. Modularidad</p> <p>Unidad 9. Comportamiento: monolito</p> <p>Unidad 10. Comportamiento: sistemas distribuidos y Big data</p> <p>Unidad 11. Disposición</p> <p>Unidad 12. Arquitectura del software y Entorno empresarial</p>	<p>Bibliografía recomendada</p>  

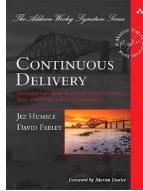
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 2: Comunicando la Arquitectura del Software</h2> <p>Comunicando la arquitectura y documentación</p> <ul style="list-style-type: none"> <li>Pautas de documentación</li> <li>Vistas y puntos de vista</li> <li>Herramientas para creación de diagramas</li> <li>Documentación y proyectos ágiles</li> <li>Directrices</li> </ul> <p>Enfoques de documentación</p> <p>Vistas, modelo C4, <b>arc42</b></p> <p>Estilo codificación arquitectónicamente evidente</p>	<p>Tema 1. Conceptos de Arquitectura de Software</p> <p><b>Unidad 1. Definiciones</b></p> <p><b>Unidad 2. Comunicando la Arquitectura del Software</b></p> <p>Unidad 3. Papel del Arquitecto del Software y Stakeholders</p> <p>Unidad 4. Atributos de calidad</p> <p>Unidad 5. Creación y validación de Arquitecturas</p> <p>Unidad 6. Técnicas de arquitectura del software</p> <p>Tema 2. Taxonomías de Arquitectura del Software</p> <p>Unidad 7. Construcción, Gestión de configuraciones</p> <p>Unidad 8. Modularidad</p> <p>Unidad 9. Comportamiento: monolito</p> <p>Unidad 10. Comportamiento: sistemas distribuidos y Big data</p> <p>Unidad 11. Disposición</p> <p>Unidad 12. Arquitectura del software y Entorno empresarial</p>	<p>Bibliografía recomendada</p>  

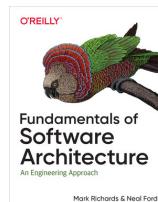
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 3: Papel del arquitecto del software y <i>stakeholders</i></h2> <p><b>Papel del arquitecto del software</b></p> <ul style="list-style-type: none"> <li>Expectativas sobre un arquitecto de software</li> <li>Aspectos de trabajo en equipo</li> <li>Topologías de equipos de trabajo</li> <li>Ley de Conway</li> </ul> <p><b>Stakeholders</b></p> <p>Identificación de <i>stakeholders</i></p>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <ul style="list-style-type: none"> <li>Unidad 1. Definiciones</li> <li>Unidad 2. Comunicando la Arquitectura del Software</li> <li><b>Unidad 3. Papel del Arquitecto del Software y Stakeholders</b></li> <li>Unidad 4. Atributos de calidad</li> <li>Unidad 5. Creación y validación de Arquitecturas</li> <li>Unidad 6. Técnicas de arquitectura del software</li> </ul> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <ul style="list-style-type: none"> <li>Unidad 7. Construcción, Gestión de configuraciones</li> <li>Unidad 8. Modularidad</li> <li>Unidad 9. Comportamiento: monolito</li> <li>Unidad 10. Comportamiento: sistemas distribuidos y Big data</li> <li>Unidad 11. Disposición</li> <li>Unidad 12. Arquitectura del software y Entorno empresarial</li> </ul>	<p><b>Bibliografía recomendada</b></p>

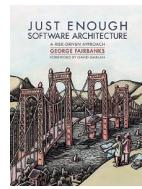
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 4: Atributos de calidad</h2> <p>Relación con requisitos funcionales</p> <p>Escenarios de calidad</p> <p>Identificación de atributos de calidad</p> <ul style="list-style-type: none"> <li>ISO25010</li> <li>Talleres de atributos de calidad</li> <li>Métricas de atributos de calidad</li> </ul>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <ul style="list-style-type: none"> <li>Unidad 1. Definiciones</li> <li>Unidad 2. Comunicando la Arquitectura del Software</li> <li><b>Unidad 3. Papel del Arquitecto del Software y Stakeholders</b></li> <li><b>Unidad 4. Atributos de calidad</b></li> <li>Unidad 5. Creación y validación de Arquitecturas</li> <li>Unidad 6. Técnicas de arquitectura del software</li> </ul> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <ul style="list-style-type: none"> <li>Unidad 7. Construcción, Gestión de configuraciones</li> <li>Unidad 8. Modularidad</li> <li>Unidad 9. Comportamiento: monolito</li> <li>Unidad 10. Comportamiento: sistemas distribuidos y Big data</li> <li>Unidad 11. Disposición</li> <li>Unidad 12. Arquitectura del software y Entorno empresarial</li> </ul>	<p><b>Bibliografía recomendada</b></p>

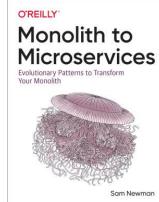
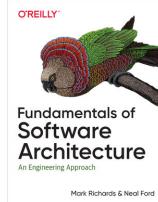
Histórico	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 5: Creación y validación</h2> <p>ADD: Attribute Driven Design Incidencias arquitectónicas Registros de decisiones arquitectónicas Evaluación de arquitecturas ATAM: Architecture Trade-off Analysis Method CBAM: Cost-benefit analysis method</p>	<p>Tema 1. Conceptos de Arquitectura de Software Unidad 1. Definiciones Unidad 2. Comunicando la Arquitectura del Software Unidad 3. Papel del Arquitecto del Software y Stakeholders Unidad 4. Atributos de calidad <b>Unidad 5. Creación y validación de arquitecturas</b> Unidad 6. Técnicas de arquitectura del software</p> <p>Tema 2. Taxonomías de Arquitectura del Software Unidad 7. Construcción, Gestión de configuraciones Unidad 8. Modularidad Unidad 9. Comportamiento: monolito Unidad 10. Comportamiento: sistemas distribuidos y Big data Unidad 11. Disposición Unidad 12. Arquitectura del software y Entorno empresarial</p>	<p>Bibliografía recomendada</p> 

Histórico	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Conceptos - unidad 6: Técnicas</h2> <p>Tácticas Estilos arquitectónicos Patrones arquitectónicos Construir vs reutilizar Arquitecturas de referencia Familias tecnológicas Frameworks COTS (<i>Commercial Off-the-shelf</i>) FOSS (<i>Free Open Source Software</i>)</p>	<p>Tema 1. Conceptos de Arquitectura de Software Unidad 1. Definiciones Unidad 2. Comunicando la Arquitectura del Software Unidad 3. Papel del Arquitecto del Software y Stakeholders Unidad 4. Atributos de calidad Unidad 5. Creación y validación de Arquitecturas <b>Unidad 6. Técnicas de arquitectura del software</b></p> <p>Tema 2. Taxonomías de Arquitectura del Software Unidad 7. Construcción, Gestión de configuraciones Unidad 8. Modularidad Unidad 9. Comportamiento: monolito Unidad 10. Comportamiento: sistemas distribuidos y Big data Unidad 11. Disposición Unidad 12. Arquitectura del software y Entorno empresarial</p>	<p>Bibliografía recomendada</p> 

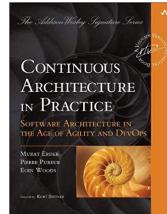
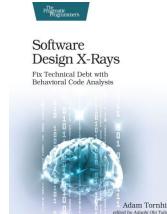
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Taxonomías - unidad 7: Construcción</h2> <p>Herramientas de construcción Gestión de configuraciones Control de versiones Versiones y <i>releases</i> Automatización de la construcción Técnicas y herramientas de trabajo en equipo Seguimiento de incidencias</p> <p>Aspectos de metodologías ágiles de desarrollo TDD Integración continua Propiedad colectiva del código</p>	<p>Tema 1. Conceptos de Arquitectura de Software Unidad 1. Definiciones Unidad 2. Comunicando la Arquitectura del Software Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i> Unidad 4. Atributos de calidad Unidad 5. Creación y validación de Arquitecturas Unidad 6. Técnicas de arquitectura del software</p> <p>Tema 2. Taxonomías de Arquitectura del Software <b>Unidad 7. Construcción, Gestión de configuraciones</b> Unidad 8. Modularidad Unidad 9. Comportamiento: monolito Unidad 10. Comportamiento: sistemas distribuidos y Big data Unidad 11. Disposición Unidad 12. Arquitectura del software y Entorno empresarial</p> <p>Bibliografía recomendada</p> 	

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Taxonomías - unidad 8: Modularidad</h2> <p>Principios de modularidad SOLID, acoplamiento, cohesividad, conocimiento...</p> <p>Estilos de modularidad Capas Aspectos Basados en dominio Domain Driven Design Arquitectura hexagonal y arquitectura limpia Modelos centrados en datos <i>Naked objects</i></p>	<p>Tema 1. Conceptos de Arquitectura de Software Unidad 1. Definiciones Unidad 2. Comunicando la Arquitectura del Software Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i> Unidad 4. Atributos de calidad Unidad 5. Creación y validación de Arquitecturas Unidad 6. Técnicas de arquitectura del software</p> <p>Tema 2. Taxonomías de Arquitectura del Software <b>Unidad 7. Construcción, Gestión de configuraciones</b> <b>Unidad 8. Modularidad</b> Unidad 9. Comportamiento: monolito Unidad 10. Comportamiento: sistemas distribuidos y Big data Unidad 11. Disposición Unidad 12. Arquitectura del software y Entorno empresarial</p> <p>Bibliografía recomendada</p>  	

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Taxonomías - Unidad 9</h2> <h3>Comportamiento: Monolito</h3> <p><b>Taxonomía de estilos arquitectónicos</b></p> <ul style="list-style-type: none"> <li>Flujos de datos (batch, pipes / filters)</li> <li>Organización trabajo (master-slave)</li> <li>Interactivos (MVC, PAC)</li> <li>Repositorio (datos compartidos, blackboard, reglas)</li> <li>Invocación (call-return, cliente-servidor, eventos)</li> <li>Sistemas adaptables (plugins, microkernel, DSLs, código móvil)</li> </ul>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <ul style="list-style-type: none"> <li>Unidad 1. Definiciones</li> <li>Unidad 2. Comunicando la Arquitectura del Software</li> <li>Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i></li> <li>Unidad 4. Atributos de calidad</li> <li>Unidad 5. Creación y validación de Arquitecturas</li> <li>Unidad 6. Técnicas de arquitectura del software</li> </ul> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <ul style="list-style-type: none"> <li>Unidad 7. Construcción, Gestión de configuraciones</li> <li>Unidad 8. Modularidad</li> <li><b>Unidad 9. Comportamiento: monolito</b></li> <li>Unidad 10. Comportamiento: sistemas distribuidos y Big data</li> <li>Unidad 11. Disposición</li> <li>Unidad 12. Arquitectura del software y Entorno empresarial</li> </ul> <p><b>Bibliografía recomendada</b></p>  	

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<h2>Taxonomías - unidad 10:</h2> <h3>Comportamiento: Sistemas distribuidos y big data</h3> <p><b>Sistemas distribuidos</b></p> <ul style="list-style-type: none"> <li>Estilos y topologías de integración</li> <li>Patrón bróker</li> <li>Peer-to-peer</li> <li>Arquitecturas orientadas a servicios</li> <li>Microservicios</li> <li>Serverless</li> </ul> <p><b>Big data</b></p> <ul style="list-style-type: none"> <li>MapReduce</li> <li>Arquitectura lambda</li> <li>Arquitectura Kappa</li> </ul>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <ul style="list-style-type: none"> <li>Unidad 1. Definiciones</li> <li>Unidad 2. Comunicando la Arquitectura del Software</li> <li>Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i></li> <li>Unidad 4. Atributos de calidad</li> <li>Unidad 5. Creación y validación de Arquitecturas</li> <li>Unidad 6. Técnicas de arquitectura del software</li> </ul> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <ul style="list-style-type: none"> <li>Unidad 7. Construcción, Gestión de configuraciones</li> <li>Unidad 8. Modularidad</li> <li><b>Unidad 9. Comportamiento: monolito</b></li> <li><b>Unidad 10. Comportamiento: sistemas distribuidos y Big data</b></li> <li>Unidad 11. Disposición</li> <li>Unidad 12. Arquitectura del software y Entorno empresarial</li> </ul> <p><b>Bibliografía recomendada</b></p>  	

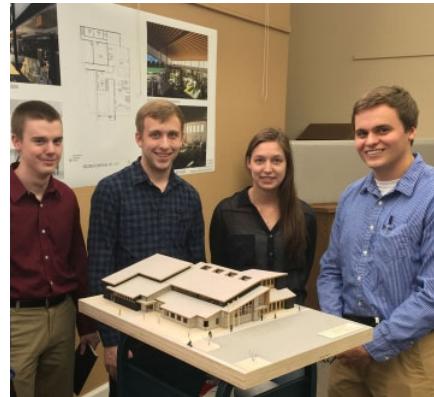
Historial	Proyecto docente - propuesta docente	Proyecto investigador
<p><b>Taxonomías - unidad 11: Disposición</b></p> <p><b>Empaquetamiento y distribución</b></p> <p>Entornos de ejecución de software</p> <p><b>Despliegue</b></p> <p>Atributos Software en producción</p> <p>Configurabilidad, observabilidad, disponibilidad</p> <p><b>Patrones para software en producción</b></p> <p>Pruebas de software en producción</p> <p>Gestión y análisis de incidencias</p>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <p>Unidad 1. Definiciones</p> <p>Unidad 2. Comunicando la Arquitectura del Software</p> <p>Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i></p> <p>Unidad 4. Atributos de calidad</p> <p>Unidad 5. Creación y validación de Arquitecturas</p> <p>Unidad 6. Técnicas de arquitectura del software</p> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <p>Unidad 7. Construcción, Gestión de configuraciones</p> <p>Unidad 8. Modularidad</p> <p>Unidad 9. Comportamiento: monolito</p> <p>Unidad 10. Comportamiento: sistemas distribuidos y Big data</p> <p><b>Unidad 11. Disposición</b></p> <p>Unidad 12. Arquitectura del software y Entorno empresarial</p>	<p>Bibliografía recomendada</p> 

Historial	Proyecto docente - propuesta docente	Proyecto investigador
<p><b>Taxonomías - unidad 12: Arquitectura del software en entorno empresarial</b></p> <p><b>Arquitectura empresarial</b></p> <p><b>Taxonomía de software empresarial</b></p> <p>ERP, CRM, SCM, WMS, ...</p> <p><b>Líneas de producto software</b></p> <p><b>Mantenimiento y evolución del software</b></p> <p>Código heredado</p> <p>Análisis de comportamiento de código</p>	<p><b>Tema 1. Conceptos de Arquitectura de Software</b></p> <p>Unidad 1. Definiciones</p> <p>Unidad 2. Comunicando la Arquitectura del Software</p> <p>Unidad 3. Papel del Arquitecto del Software y <i>Stakeholders</i></p> <p>Unidad 4. Atributos de calidad</p> <p>Unidad 5. Creación y validación de Arquitecturas</p> <p>Unidad 6. Técnicas de arquitectura del software</p> <p><b>Tema 2. Taxonomías de Arquitectura del Software</b></p> <p>Unidad 7. Construcción, Gestión de configuraciones</p> <p>Unidad 8. Modularidad</p> <p>Unidad 9. Comportamiento: monolito</p> <p>Unidad 10. Comportamiento: sistemas distribuidos y Big data</p> <p><b>Unidad 11. Disposición</b></p> <p><b>Unidad 12. Arquitectura del software y Entorno empresarial</b></p>	<p>Bibliografía recomendada</p>  

Historial	Proyecto docente - propuesta docente							Proyecto investigador																																																						
Organización de la asignatura en horas																																																														
150h totales = 60h presenciales + 90h no presenciales																																																														
21h expositivas + 7h seminarios + 28h práct. Lab. + 2 tut. Grupales + 2 eval.																																																														
<table border="1"> <thead> <tr> <th rowspan="2">Temas</th> <th rowspan="2">Horas totales</th> <th colspan="6">Actividades presenciales</th> <th colspan="3">No presenciales</th> </tr> <tr> <th>Clase Expositiva</th> <th>Seminarios</th> <th>Prácticas de laboratorio</th> <th>Tutorías grupales</th> <th>Sesiones de Evaluación</th> <th>Total</th> <th>Trabajo grupo</th> <th>Trabajo autónomo</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1 (Conceptos)</td> <td>40</td> <td>7</td> <td>2</td> <td>4</td> <td>1</td> <td>1</td> <td>15</td> <td>15</td> <td>10</td> <td>25</td> </tr> <tr> <td>2 (Taxonomías)</td> <td>110</td> <td>14</td> <td>5</td> <td>24</td> <td>1</td> <td>1</td> <td>45</td> <td>45</td> <td>20</td> <td>65</td> </tr> <tr> <td>Total</td> <td>150</td> <td>21</td> <td>7</td> <td>28</td> <td>2</td> <td>2</td> <td>60</td> <td>60</td> <td>30</td> <td>90</td> </tr> </tbody> </table>										Temas	Horas totales	Actividades presenciales						No presenciales			Clase Expositiva	Seminarios	Prácticas de laboratorio	Tutorías grupales	Sesiones de Evaluación	Total	Trabajo grupo	Trabajo autónomo	Total	1 (Conceptos)	40	7	2	4	1	1	15	15	10	25	2 (Taxonomías)	110	14	5	24	1	1	45	45	20	65	Total	150	21	7	28	2	2	60	60	30	90
Temas	Horas totales	Actividades presenciales						No presenciales																																																						
		Clase Expositiva	Seminarios	Prácticas de laboratorio	Tutorías grupales	Sesiones de Evaluación	Total	Trabajo grupo	Trabajo autónomo	Total																																																				
1 (Conceptos)	40	7	2	4	1	1	15	15	10	25																																																				
2 (Taxonomías)	110	14	5	24	1	1	45	45	20	65																																																				
Total	150	21	7	28	2	2	60	60	30	90																																																				

Historial	Proyecto docente - propuesta docente				Proyecto investigador				
Clases expositivas y conferencias									
Clases en aula grande									
Se organizan unas 2 conferencias cada curso									
PONENTE	CARGO	EMPRESA	TÍTULO	FECHA					
Jorge Manrubia Díez	Prog. senior	BaseCamp	Luchando contra los mercaderes de la complejidad	14/Abr/2021					
Ángel Maldonado	CEO	Empathy.co	Descentralización abierta	10/Feb/2021					
Diego Berrueta	Principal engineer	Atlassian	Escalando hacia miles de microservicios	1/Abr/2020					
Alejandro Juan García	Project manager	Empathy.co	Búsqueda en SOLID	19/Feb/2020					
Pablo Bermejo	Eng. manager	DXC Techn.	Arquitectura del software #10YearsChallenge	20/Mar/2019					
Sergio Fernández	Soft. Engineer	Amazon	Tendencias en arquitectura del software	30/Ene/2019					
Luis Fernández Palacio	Arq. Software	CapGemini	Arquitectura del software basada en microservicios	9/Abr/2018					
Manuel Jesús Fernández	Quality Manager	Mecalux	Arquitectura del software Mecalux EasyWMS	20/Abr/2017					
Miguel Fernández	Senior engineer	Github	Refactorizando sistemas a escala	21/Mar/2017					

Historial	Proyecto docente - propuesta docente			Proyecto investigador			
<h2>Seminarios</h2>							
<b>Desde 2017-18: Presentaciones realizadas por estudiantes</b>							
Inicialmente, sobre proyectos reales de código abierto							
Curso pasado, sobre temas relacionados con arquitectura del software							
Maven	Asciidoc	PlantUML	Architecture decision records				
Spring boot	Pandoc	Modelio	Agile architecture				
Android studio	Dokieli	Yarn	Continuous delivery				
Ionic	Mocha	npx	Branching patterns				
Angular	WebRTC	Jest	Clean architecture				
Apache zookeeper	WebPack	Fast-check	Event sourcing				
Akka	Varnish	React-native	Microservices trade-offs				
Gatling	Memcached	Redux	Circuit breaker				
Apache Mesos	IPFS	Redis	Chaos engineering				
Docker	gitter	gRPC	Serverless				
Kubernetes							
Apache Spark							
Apache flink							
2017-18	Lagom framework	2018-19	2019-20	2020-21			

Historial	Proyecto docente - propuesta docente			Proyecto investigador			
<h2>Prácticas de laboratorio</h2>							
<b>Trabajo en grupo a lo largo del curso</b>							
Plataforma github							
<b>Todos los grupos resuelven mismo tema</b>							
<b>Trabajo realizado en grupo</b>							
Creación/documentación de una arquitectura							
Implementación de un prototipo							
							

Historial	<b>Proyecto docente - propuesta docente</b>	Proyecto investigador
<h2>Sesiones de control a lo largo del curso</h2> <p><b>1er sesión de control - Semana 5</b>  <a href="#">Documentación 0.1</a></p> <p><b>2º sesión de control - Semana 8</b>  <a href="#">Prototipo versión 0.1</a></p> <p><b>3er sesión de control - Semana 11</b>  <a href="#">Prototipo versión 1.0 + Documentación 1.0</a></p> <p><b>4º sesión de control - Semana 14</b>  <a href="#">Prototipo versión 1.1 + Documentación 1.1</a>  <a href="#">Presentación pública</a></p>		

Historial	<b>Proyecto docente - propuesta docente</b>	Proyecto investigador		
<h2>Estructura de sesiones prácticas</h2> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; vertical-align: top; padding: 5px;"> <b>Jornadas normales</b>            1h explicación            1h trabajo grupo            (deben tomar acta)         </td> <td style="width: 85%; vertical-align: top; padding: 5px;">           1 Introducción a la práctica, organización de equipos            2 Control de versiones: git y github            3 Diagramas UML de arquitectura (diagramas de clase, componentes, despliegue, secuencia). Herramientas para creación de diagramas UML.            Introducción a documentación y plantillas arc42            4 Introducción a tecnologías para la práctica: React y Solid  <b>5 1er sesión de control: documentación</b>            6 Automatización de la construcción: Maven, Gradle, npm, etc.            7 Distribución y despliegue: páginas github, docker  <b>8 2º sesión control: primer prototipo</b>            9 Desarrollo basado en pruebas (TDD) e integración continua            10 Pruebas de aceptación y desarrollo basado en comportamiento (BDD)  <b>11 3er sesión control: Primera release con documentación actualizada</b>            12 Pruebas de carga y otras pruebas: usabilidad, seguridad, escalabilidad, etc.            13 Monitorización y evaluación de rendimiento (<i>profiling</i>)  <b>14 Día de la presentación y release final</b> </td> </tr> </table>			<b>Jornadas normales</b> 1h explicación 1h trabajo grupo (deben tomar acta)	1 Introducción a la práctica, organización de equipos 2 Control de versiones: git y github 3 Diagramas UML de arquitectura (diagramas de clase, componentes, despliegue, secuencia). Herramientas para creación de diagramas UML. Introducción a documentación y plantillas arc42 4 Introducción a tecnologías para la práctica: React y Solid <b>5 1er sesión de control: documentación</b> 6 Automatización de la construcción: Maven, Gradle, npm, etc. 7 Distribución y despliegue: páginas github, docker <b>8 2º sesión control: primer prototipo</b> 9 Desarrollo basado en pruebas (TDD) e integración continua 10 Pruebas de aceptación y desarrollo basado en comportamiento (BDD) <b>11 3er sesión control: Primera release con documentación actualizada</b> 12 Pruebas de carga y otras pruebas: usabilidad, seguridad, escalabilidad, etc. 13 Monitorización y evaluación de rendimiento ( <i>profiling</i> ) <b>14 Día de la presentación y release final</b>
<b>Jornadas normales</b> 1h explicación 1h trabajo grupo (deben tomar acta)	1 Introducción a la práctica, organización de equipos 2 Control de versiones: git y github 3 Diagramas UML de arquitectura (diagramas de clase, componentes, despliegue, secuencia). Herramientas para creación de diagramas UML. Introducción a documentación y plantillas arc42 4 Introducción a tecnologías para la práctica: React y Solid <b>5 1er sesión de control: documentación</b> 6 Automatización de la construcción: Maven, Gradle, npm, etc. 7 Distribución y despliegue: páginas github, docker <b>8 2º sesión control: primer prototipo</b> 9 Desarrollo basado en pruebas (TDD) e integración continua 10 Pruebas de aceptación y desarrollo basado en comportamiento (BDD) <b>11 3er sesión control: Primera release con documentación actualizada</b> 12 Pruebas de carga y otras pruebas: usabilidad, seguridad, escalabilidad, etc. 13 Monitorización y evaluación de rendimiento ( <i>profiling</i> ) <b>14 Día de la presentación y release final</b>			

[Historial](#)[Proyecto docente - propuesta docente](#)[Proyecto investigador](#)

## Presentación final

Acto público (Salón de actos)

Los profesores de la asignatura actúan como jurado

Los grupos presentan sus trabajos (arquitectura + prototipo)

[Historial](#)[Proyecto docente - propuesta docente](#)[Proyecto investigador](#)

## Concurso "Solid World Challenge"

Comenzó a realizarse en curso 2018-19

Jurado

Profesor de la Escuela

Miembro de Inrupt

Miembro de Empathy

Premio al mejor equipo:

Diploma firmado por Tim Berners-Lee

Oferta de beca/contrato en formación por empresa Empathy



Historial		Proyecto docente - propuesta docente	Proyecto investigador																																								
<h2>Temas de prácticas</h2> <p><b>Temas abiertos y novedosos</b></p> <p><b>Utilización de tecnologías complejas (librerías externas)</b></p> <table border="1"> <thead> <tr> <th>Curso</th> <th>Nombre</th> <th>Breve descripción</th> <th>Observaciones/tecnologías</th> </tr> </thead> <tbody> <tr> <td>2020-21</td> <td>radarin</td> <td>Radar de amigos cercanos</td> <td>Tecnologías SOLID + móvil + servidor</td> </tr> <tr> <td>2019-20</td> <td>viade</td> <td>Rutas descentralizadas</td> <td>Tecnologías SOLID + GPS</td> </tr> <tr> <td>2018-19</td> <td>dechat</td> <td>Chat descentralizado</td> <td>Tecnologías SOLID</td> </tr> <tr> <td>2017-18</td> <td>inciManager</td> <td>Gestión de incidencias ciudadanos</td> <td>Tecnologías Apache Kafka</td> </tr> <tr> <td>2016-17</td> <td>Citizens</td> <td>Sistema de participación ciudadana</td> <td>Tecnologías Apache Kafka</td> </tr> <tr> <td>2015-16</td> <td>VotingSystem</td> <td>Sistema de votaciones digitales</td> <td>Spring boot + websockets</td> </tr> <tr> <td>2014-15</td> <td>Trivial</td> <td>Juego trivial en Web</td> <td>Spring boot + websockets</td> </tr> <tr> <td>2013-14</td> <td>ObservaTerra</td> <td>Gestión de observaciones sobre la tierra</td> <td>Tecnologías libres (Play! Framework)</td> </tr> <tr> <td>2012-13</td> <td>Entrecine</td> <td>Gestión web de un cine</td> <td>Tecnologías libres (Play! Framework)</td> </tr> </tbody> </table>				Curso	Nombre	Breve descripción	Observaciones/tecnologías	2020-21	radarin	Radar de amigos cercanos	Tecnologías SOLID + móvil + servidor	2019-20	viade	Rutas descentralizadas	Tecnologías SOLID + GPS	2018-19	dechat	Chat descentralizado	Tecnologías SOLID	2017-18	inciManager	Gestión de incidencias ciudadanos	Tecnologías Apache Kafka	2016-17	Citizens	Sistema de participación ciudadana	Tecnologías Apache Kafka	2015-16	VotingSystem	Sistema de votaciones digitales	Spring boot + websockets	2014-15	Trivial	Juego trivial en Web	Spring boot + websockets	2013-14	ObservaTerra	Gestión de observaciones sobre la tierra	Tecnologías libres (Play! Framework)	2012-13	Entrecine	Gestión web de un cine	Tecnologías libres (Play! Framework)
Curso	Nombre	Breve descripción	Observaciones/tecnologías																																								
2020-21	radarin	Radar de amigos cercanos	Tecnologías SOLID + móvil + servidor																																								
2019-20	viade	Rutas descentralizadas	Tecnologías SOLID + GPS																																								
2018-19	dechat	Chat descentralizado	Tecnologías SOLID																																								
2017-18	inciManager	Gestión de incidencias ciudadanos	Tecnologías Apache Kafka																																								
2016-17	Citizens	Sistema de participación ciudadana	Tecnologías Apache Kafka																																								
2015-16	VotingSystem	Sistema de votaciones digitales	Spring boot + websockets																																								
2014-15	Trivial	Juego trivial en Web	Spring boot + websockets																																								
2013-14	ObservaTerra	Gestión de observaciones sobre la tierra	Tecnologías libres (Play! Framework)																																								
2012-13	Entrecine	Gestión web de un cine	Tecnologías libres (Play! Framework)																																								

Historial		Proyecto docente - propuesta docente	Proyecto investigador
<h2>Evaluación</h2> <p><b>Teoría (40%)</b></p> <p>Examen final (prueba objetiva) - 70%</p> <p>Pruebas a lo largo del curso - 30%</p> <p><b>Práctica (60%)</b></p> <p>Trabajo en grupo - 70%</p> <p>Trabajo individual - 30%</p> <p>Analizando contribuciones en <i>github</i></p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <math display="block">\text{Final} = \text{Teoría} \times 40\% + \text{Práctica} \times 60\%</math> <p>donde:</p> <math display="block">\text{Teoría} = \text{Examen} \times 70\% + \text{Seminarios} \times 30\%</math> <math display="block">\text{Práctica} = \text{Grupo} \times 70\% + \text{Individual} \times 30\%</math> </div>			

[Historial](#)

[Proyecto docente - propuesta docente](#)

[Proyecto investigador](#)

# Recursos docentes

Página Web

Página Moodle Campus Virtual

Teams

Github

Zapiens/Kahoot

- Historial
- Proyecto docente - propuesta docente
- Proyecto investigador

## Posibles variaciones

- Involucrar *mantenibilidad* en la práctica
  - Aspectos de código heredado (*legacy*)
- Granularidad de las prácticas
  - Prácticas sobre diferentes estilos arquitectónicos
  - Prácticas individuales o en grupos más pequeños
- Integración de práctica con otras asignaturas
  - Ejemplo: Dirección y planificación de proyectos
- Mover asignatura de semestre
  - Podrían utilizarse conceptos de ingeniería de requisitos
- Enseñanza no presencial
  - Experimento ya realizado y superado

Historial | Proyecto docente - propuesta docente | Proyecto investigador

## Recomendaciones bibliográficas

The books shown in the grid include:

- Software Architecture in Practice, Third Edition
- Documenting Software Architectures, Second Edition
- Fundamentals of Software Architecture: An Engineering Approach
- The Software Architect Elevator: Building the Architect's Role in the Digital Enterprise
- JUST ENOUGH SOFTWARE ARCHITECTURE: A Practical Guide to Managing Complexity in the Digital Enterprise
- Designing Software Architectures: A Practical Approach
- CONTINUOUS DELIVERY: Reliable Software Releases Through Lean Practices and Infrastructure
- Monolith to Microservices: Evolutionary Patterns in Transition
- Software Systems Architecture, Second Edition: Working With Stakeholders Using Models and Pictures
- CONTINUOUS ARCHITECTURE IN PRACTICE: Software Architecture in the Age of Agile and DevOps
- Building Evolutionary Architectures
- Domain-Driven DESIGN: Tackling Complexity in the Heart of Software
- Release It!: Design and Deploy Production-Ready Software
- Design It!: The Pragmatic Guide to Better Architectures
- Clean Architecture: A Craftsman's Guide to Software Structure and Design
- Software Engineering at Google
- TEAM TOPOLOGIES: ORGANIZING YOUR BUSINESS AROUND YOUR FLUID TEAM FLOW

## Parte III Proyecto de investigación

Jose Emilio Labra Gayo



Universidad de Oviedo

Historial | Proyecto docente | **Proyecto investigación**



Applying kNowledge Graphs to research data interoperabiLlity and ReUsability (ANGLIRU)

Historial | Proyecto docente | **Proyecto investigación**

## Datos del proyecto

**Tipo de proyecto:** Coordinado

**IP subproyecto 1:** *Jose Emilio Labra Gayo* (Univ. de Oviedo)  
Applying kNowledge Graphs to research data interoperabiLlity ([ANGLI](#))

**IP subproyecto 2:** *Roberto García González* (Univ. de Lleida)  
Applying kNowledge Graphs to research data ReUsability ([ANGRU](#))

**Convocatoria:** Tipo B (Retos Investigación), Plan Estatal I+D+i (2020-2023)

**Reto principal:** 7º Economía, sociedad y cultura digitales

**Otro reto:** 6º Ciencias sociales y humanidades y ciencia con y para la sociedad

## Objetivos principales

Mejorar gestión de datos de investigación siguiendo **FAIR**

### *F*indability y *A*ccessibility

Generación e integración con grafos de conocimiento sobre investigación

### *I*nteroperability

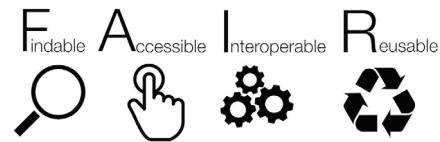
Integración y creación de modelos de datos

Calidad y verificación de datos

### *R*eusability

Gestión semántica de licencias

Procedencia de datos y confianza

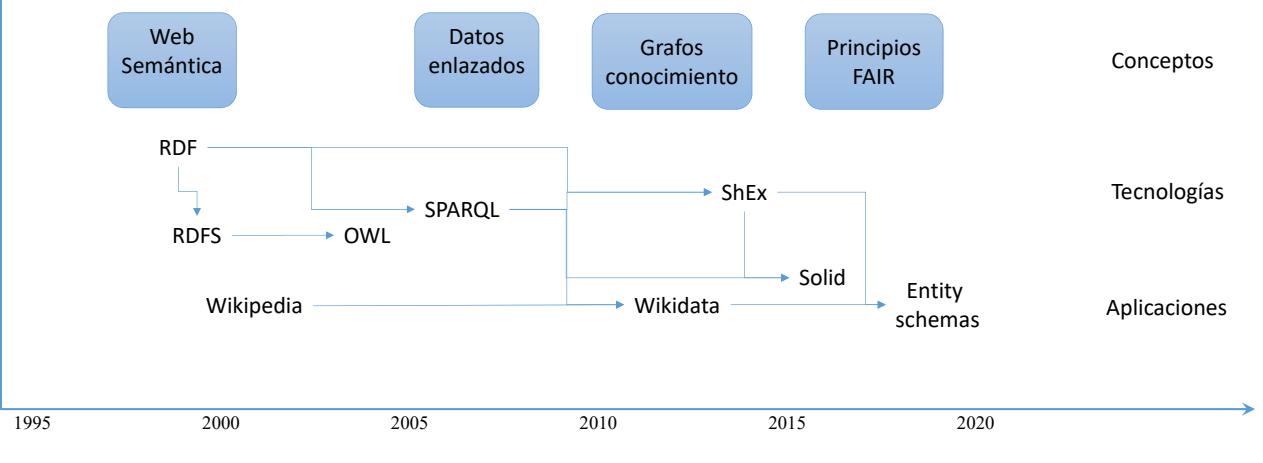


### Casos de uso

Bioinformática y Salud

Agricultura

## Antecedentes



Historial	Proyecto docente	Proyecto investigación
<h2>Trabajos previos del equipo</h2>  		
<p>Experiencia en Web semántica, datos enlazados, RDF</p> <p>Creación de ShEx/SHACL y herramientas de modelado de datos</p> <p>Participación Proyecto Hércules, ontología investigación</p> <p>Recientes colaboraciones utilizando Wikidata en el ámbito de la bioinformática y salud</p> <p>Contratos y consultorías sobre tecnologías semánticas y grafos de conocimiento</p>	<p>Implantación sistema de gestión de investigación <i>Dataverse</i> en UdL</p> <p>Participación 2 proyectos nacionales utilizando datos semánticos</p> <p>Aplicación grafos conocimiento a sector de agricultura</p> <p>Participación proyecto europeo Invid - gestión licencias y <i>blockchain</i></p> <p>Contratos y consultorías sobre grafos de conocimiento con empresas como <i>Sony</i>, <i>Cambridge Semantics</i>, etc.</p>	<p>Estancia de investigación de Roberto García y Roga Gil en WESO, verano 2020</p>

Historial	Proyecto docente	Proyecto investigación
<h2>Objetivos específicos - WESO</h2> 		
<ul style="list-style-type: none"> <li>• SO1 Transformación entre fuentes y formatos de datos heterogéneos como XML, CSV ó JSON a RDF</li> <li>• SO2 Validación a gran escala de grafos de conocimiento</li> <li>• SO3 Extracción de modelos de datos y esquemas a partir de datos RDF existentes</li> <li>• SO4 Creación de shapes de datos RDF, visualización y mapeos.</li> <li>• SO5 Investigación sobre la interacción entre ontologías, vocabularios, reglas y <i>shapes</i></li> <li>• SO6 Interoperabilidad entre datos de investigación y almacenes de datos personales</li> <li>• SO7 Representación FAIR de datos clínicos representados como FHIR.</li> <li>• SO8 Subconjuntos de grafos de conocimiento en dominios específicos como las ciencias biológicas y de la salud</li> <li>• SO9 Chequeo de hechos basado en grafos de conocimiento sobre información de salud</li> </ul>		

Historial	Proyecto docente	Proyecto investigación
-----------	------------------	------------------------

## Objetivos específicos - UdL



SO10: Integración de ANGLIRU con repositorios de datos de investigación existentes

SO11: Generación de grafos de conocimiento a partir de datos y metadatos recuperados de almacenes de datos de investigación

SO12: Enriquecimiento de grafos de conocimiento mediante anotaciones semi-automáticas para mejorar la calidad de datos, el descubrimiento y la integración.

SO13: Mejora de la concienciación de los reutilizadores sobre la estructura de los conjuntos de datos y las oportunidades de reutilización

SO14: Contribuir a las oportunidades de creación de licencias de conjuntos de datos ampliando el rango de licencias mediante representaciones que faciliten la gestión automatizada y la confianza de las partes

SO15 Facilitar la reutilización de datos mediante extracciones de datos enfocadas a las herramientas utilizadas por los investigadores, incluidos modelos para aplicar métodos de inteligencia artificial

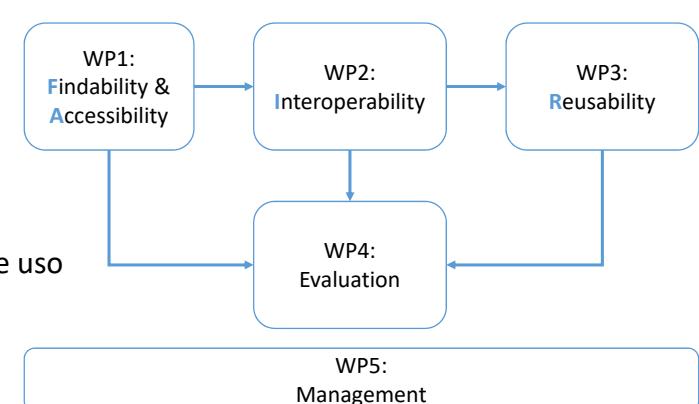
SO16 Aplicar grafos de conocimiento en el contexto de datos de investigación sobre agricultura

Historial	Proyecto docente	Proyecto investigación
-----------	------------------	------------------------

## Plan de trabajo

**5 paquetes de trabajo**

- WP1. **Findability & Accessibility**
- WP2. **Interoperability**
- WP3. **Reusability**
- WP4. Evaluación mediante casos de uso
  - Agricultura y pacto verde
  - Salud y bioinformática
- WP5. Gestión
  - Gestión del proyecto
  - Diseminación



```

graph LR
    WP1[WP1: Findability & Accessibility] --> WP2[WP2: Interoperability]
    WP2 --> WP3[WP3: Reusability]
    WP1 --> WP4[WP4: Evaluation]
    WP3 --> WP4
    WP4 --> WP5[WP5: Management]
  
```

**Historial** | **Proyecto docente** | **Proyecto investigación**

## Equipo de investigación/trabajo

					
Jose E. Labra G. (JELG)	Patricia Ordóñez (POP)	Daniel Gayo A. (DGA)	Roberto García G. (RGG)	Rosa Gil I. (RGI)	Marta Oliva S. (MOS)
					
Daniel Fernández A. (DFA)	WESO	Aquilino A. Juan (AAJF)	Eva Estupinyà P. (ESP)	Juan M. Gimeno (JGI)	
 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>
Doctorando	Doctorando	Desarrollador	Desarrollador	Doctorando	Doctorando
<b>Personal propuesto para contratar</b>					
 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>	 <b>WESO</b>		
Desarrollador	Desarrollador	Desarrollador	Desarrollador		

**Historial** | **Proyecto docente** | **Proyecto investigación**

## Cronograma: tareas

ID	Tarea	Líder	Año 1				Año 2				Año 3			
			Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
T1.1	Integración repositorios de investigación existentes	EEP												
T1.2	Generación dataset de grafos de conocimiento	MOS												
T1.3	Enriquecimiento semántico de grafos de conocim.	RGG												
T2.1	Modelos de datos y validación de grafos de con.	JELG												
T2.2	Creación, visualización y mapeo de esquemas	POP												
T2.3	Vocabularios, shapes and reglas para grafos de con.	DGA												
T2.4	Interoperabilidad datos inv. públicos y privados	AAJF												
T3.1	Servicio descubrimiento estructura dataset	JGI												
T3.2	Gestión de licencias de reutilización	JGI												
T3.3	Confianza en reutilización	RGG												
T3.4	Extracción de datos	RGG												
T4.1	Agricultura y Pacto Verde	RGG												
T4.2	Salud y bioinformática	DGA												
T5.1	Gestión del Proyecto	POP												
T5.2	Diseminación	JELG												

		Historial	Proyecto docente				Proyecto investigación							
ID	Entregables	Líder	Año 1				Año 2				Año 3			
			Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
D1.1	Integración ANGLIRU y Dataverse	ESP		■										
D1.2	Versión preliminar conversor Dataset a Grafo de Con.	MOS			■									
D1.3	Versión preliminar enriquecedor Grafo de Con.	RGG				■								
D1.4	Integración ANGLIRU y Zenodo	ESP					■							
D1.5	Versión final conversor Dataset a Grafo de Con.	MOS						■						
D1.6	Versión final enriquecedor Grafo de Con.	RGG							■					
D2.1	Servicio transformación de datos	JELG												
D2.2	Servicio validación a gran escala	JELG												
D2.3	Servicio extracción de esquemas a partir de datos	DGA					■							
D2.4	Servicio creación y visualización de esquemas	POP				■								
D2.5	Servicio mapeado entre esquemas	AAJF									■			
D2.6	Servicio integración continua basado en Shapes	JELG									■			
D2.7	Grafo de con. Con shapes, reglas y ontologías	DGA												■
D2.8	Servicio interoperabilidad con SOLID	AAJF												■
D3.1	Servicio descubrimiento Dataset	JGI							■					
D3.2	Servicio licenciamiento Dataset	RGG							■					
D3.3	Servicio reutilización de confianza	RGG									■			
D3.4	Servicio extracción de datos	RGG										■		
D4.1	Prototipo Grafo Con. Agricultura y Pacto Verde	JGI									■			
D4.2	Grafo Con. Agricultura y Pacto Verde	RGG												■
D4.3	Servicio creación de subconjuntos para GeneWiki	RGG									■			
D4.4	Servicio chequeo hechos en Wikidata sobre salud	RGG												■
D5.1	Plan gestión de riesgos	POP	■											
D5.2	Revisión Plan Gestión de Datos	EEP	■					■			■			
D5.3	Página Web sobre proyecto	POP	■											

		Historial	Proyecto docente				Proyecto investigación						
<h2>Impacto y resultados esperados</h2>													
<p>Mejorar cumplimiento principios FAIR en plataformas de gestión de investigación</p>													
<p>Integración mediante servicios</p>													
<p>Mejorar interoperabilidad y calidad de datos</p>													
<p>Herramientas para creación de modelos de datos semánticos y validación</p>													
<p>Mejorar reusabilidad y gestión de datos</p>													
<p>Gestión de licencias y confianza</p>													
<p>Mejorar datos de investigación sobre</p>													
<p>Agricultura y Pacto Verde</p>													
<p>Salud y bioinformática</p>													

Historial	Proyecto docente	Proyecto investigación																		
<h2>Notificación</h2>																				
<p>Proyecto aprobado y propuesto para financiación</p>																				
<p>Periodo: 2021-2023</p>																				
<p>Cantidad: 65.703€ (subproyecto 1) + 77.924€ (subproyecto 2)</p>																				
<table border="1"><tbody><tr><td>230</td><td>PID2020-117912RB-C21</td><td>65.703,00</td><td>54.300,00</td><td>11.403,00</td><td>10.512,48</td><td>17.082,78</td><td>38.107,74</td><td>0,00</td></tr><tr><td>231</td><td>PID2020-117912RB-C22</td><td>77.924,00</td><td>64.400,00</td><td>13.524,00</td><td>12.467,84</td><td>20.260,24</td><td>45.195,92</td><td>0,00</td></tr></tbody></table>			230	PID2020-117912RB-C21	65.703,00	54.300,00	11.403,00	10.512,48	17.082,78	38.107,74	0,00	231	PID2020-117912RB-C22	77.924,00	64.400,00	13.524,00	12.467,84	20.260,24	45.195,92	0,00
230	PID2020-117912RB-C21	65.703,00	54.300,00	11.403,00	10.512,48	17.082,78	38.107,74	0,00												
231	PID2020-117912RB-C22	77.924,00	64.400,00	13.524,00	12.467,84	20.260,24	45.195,92	0,00												

Fin presentación 1<sup>a</sup> prueba

Jose Emilio Labra Gayo



Universidad de Oviedo

# Segunda prueba Trabajo de investigación

Jose Emilio Labra Gayo



Universidad de Oviedo

# Creación de subconjuntos de grafos de conocimiento mediante Shape Expressions

Jose Emilio Labra Gayo



## Razones para elección del tema

Continúa investigación sobre *Shape Expressions* y sus aplicaciones

Resolver problema práctico, propuesto por la comunidad

Participación en biohackathones con esa temática

Virtual Biohackathon 2020

SWAT4(HC)LS Hackathon 2021

Adopción de *Shape Expressions* por Wikidata

Contrato "*Robustifying Scholia*", Univ. Virginia, *Sloan Foundation*

Interés de Andra Waagmeester y proyecto GeneWiki

Parte de entregable proyecto ANGLIRU

Avance en la disciplina: Mejorar/facilitar consumo de datos semánticos

## Principales resultados del artículo

Definición formal de grafos RDF, *property graphs* y *wikibase graphs*

Extensión de ShEx para *property graphs* (PShEx)

Extensión de ShEx para *wikibase graphs* (WShEx)

Definición de 5 técnicas de generación de subconjuntos

Entity-matching

Simple matching

ShEx-based matching

Shex+Slurp

ShEx+Pregel

Definición e implementación algoritmo validación a gran escala sobre Spark GraphX

# Formato del trabajo

## Artículo científico en inglés

La convocatoria no indica requisitos de idioma

El inglés permite su posterior presentación en revista/congreso

Publicado en abierto como preprint en Arxiv

<https://arxiv.org/abs/2110.11709>

Objetivos a corto plazo:

Optimizar implementaciones

Terminar experimentos

Enviar a publicación

The screenshot shows the Arxiv preprint page for the paper 'Creating Knowledge Graphs Subsets using Shape Expressions' by Jose Emilio Labra Gayo. The page includes the Arxiv logo, Cornell University logo, search bar, and navigation links. The main content area displays the abstract, author information, and the full text of the paper. The text discusses the challenges of creating subsets of knowledge graphs and proposes a novel validation algorithm for SHEx based on the Preger algorithm.

# Creating Knowledge Graphs subsets using Shape Expressions

Jose Emilio Labra Gayo



## Motivation and main idea (1)

Since its proposal in 2012, KGs are very successful

Different kinds of KGs

Free/open and collaborative like Wikidata

Enterprise/proprietary and owned by companies:  
Google, Amazon, eBay, Facebook, Microsoft,...

Size of their contents is continually growing

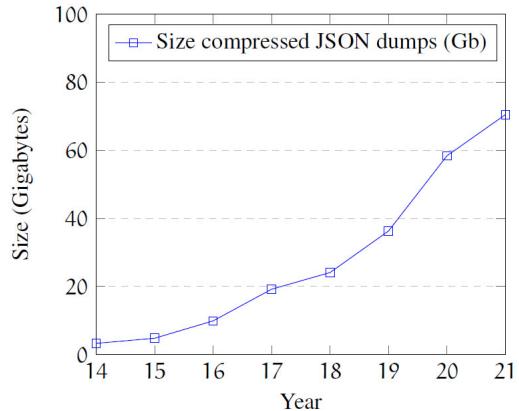
Problem: KGs victims of their own success?

Example: size of Wikidata dumps almost doubling

30GB (2014 - uncompressed)

1.256GB (2021 - uncompressed)

Difficult to produce/consume/handle KGs



Idea 1: create subset of KGs for some domain



## Motivation and main idea (2)

Shape Expressions: language to describe and validate RDF

Concise and human-friendly

Automatically processable

Several tools already exist around ShEx

Target audience: domain data experts

Already adopted by Wikidata to validate RDF serialization of Wikibase entities

```

<Researcher> {
  :name      xsd:string ;
  :birthDate xsd:date ? ;
  :birthPlace @<Place> ? ;
  :knows     @<Researcher> *
}

<Place> {
  :name      xsd:string ;
  :country   @<Country>
}

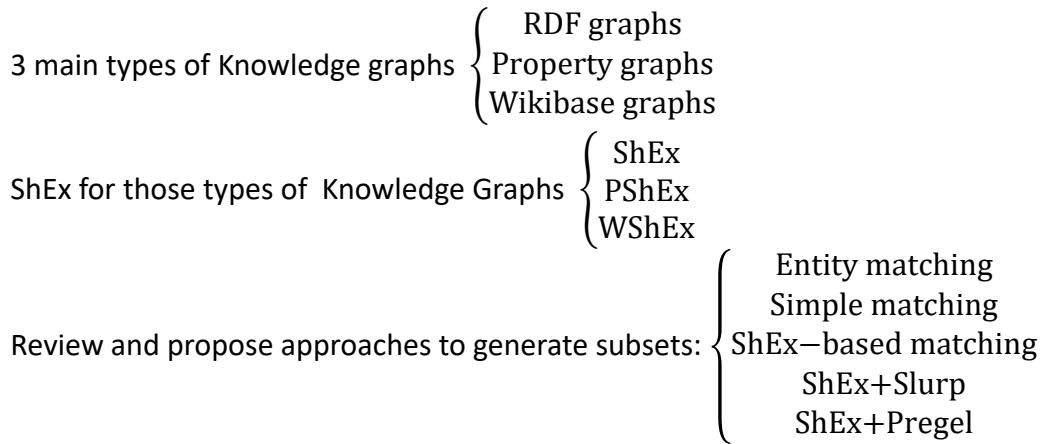
<Country> {
  :name      xsd:string ;
}
  
```

Example of a ShEx schema

Idea 2: use Shape expressions to define those subsets



## Structure of the presentation



## RDF graphs

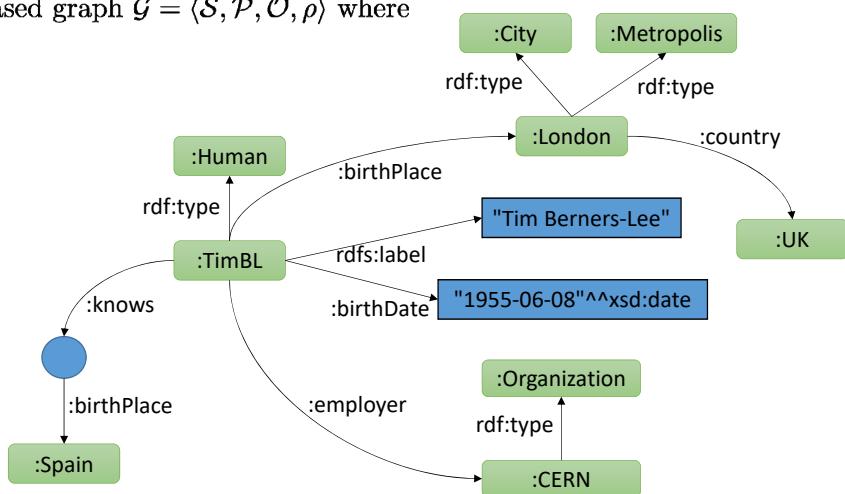
Given a set of IRIs  $\mathcal{I}$ , a set of blank nodes  $\mathcal{B}$  and a set of literals  $\text{Lit}$   
 an *RDF graph* is a triple based graph  $\mathcal{G} = \langle \mathcal{S}, \mathcal{P}, \mathcal{O}, \rho \rangle$  where

$$\mathcal{S} = \mathcal{I} \cup \mathcal{B},$$

$$\mathcal{P} = \mathcal{I},$$

$$\mathcal{O} = \mathcal{I} \cup \mathcal{B} \cup \text{Lit}$$

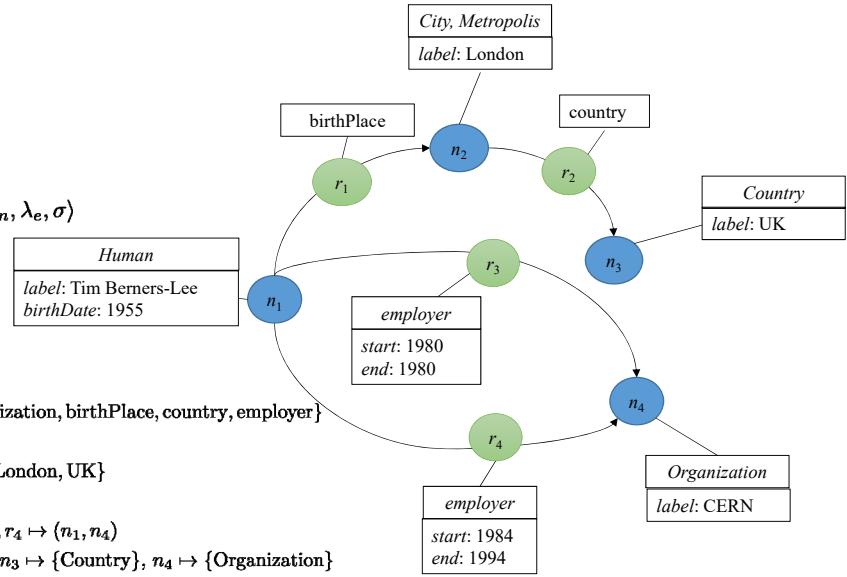
$$\rho \subseteq \mathcal{S} \times \mathcal{P} \times \mathcal{O}$$



## Property graphs

Given a set of types  $\mathcal{T}$ ,  
 a set of properties  $\mathcal{P}$ ,  
 and a set of values  $\mathcal{V}$ ,  
 a *property graph*  $\mathcal{G}$  is a tuple  $\langle \mathcal{N}, \mathcal{E}, \rho, \lambda_n, \lambda_e, \sigma \rangle$   
 where  $\mathcal{N} \cap \mathcal{E} = \emptyset$ ,  
 $\rho : \mathcal{E} \mapsto \mathcal{N} \times \mathcal{N}$  is a total function  
 $\lambda_n : \mathcal{N} \mapsto \text{FinSet}(\mathcal{T})$   
 $\lambda_e : \mathcal{E} \mapsto \mathcal{T}$   
 $\sigma : \mathcal{N} \cup \mathcal{E} \times \mathcal{P} \mapsto \text{FinSet}(\mathcal{V})$

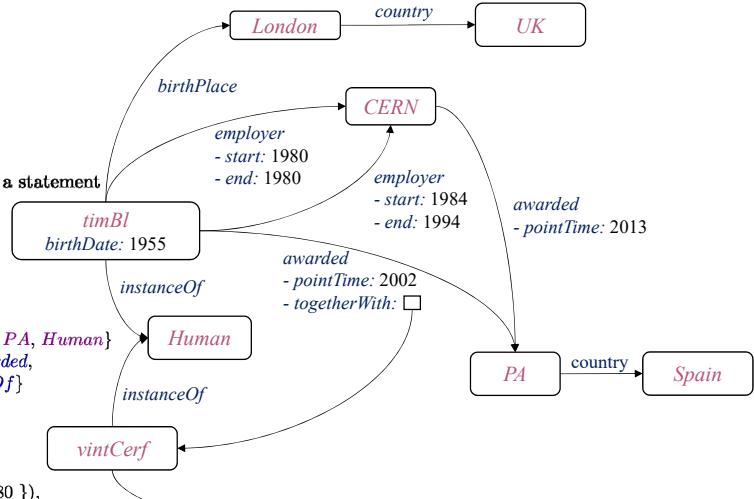
$\mathcal{T} = \{\text{Human, City, Metropolis, Country, Organization, birthPlace, country, employer}\}$   
 $\mathcal{P} = \{\text{label, birthDate, start, end}\}$   
 $\mathcal{V} = \{\text{Tim Berners-Lee, 1955, 1980, 1984, 1994, London, UK}\}$   
 $\mathcal{N} = \{n_1, n_2, n_3, n_4\} \quad \mathcal{E} = \{r_1, r_2, r_3, r_4\}$   
 $\rho = r_1 \mapsto (n_1, n_2), r_2 \mapsto (n_2, n_3), r_3 \mapsto (n_1, n_4), r_4 \mapsto (n_1, n_4)$   
 $\lambda_n = n_1 \mapsto \{\text{Human}\}, n_2 \mapsto \{\text{City, Metropolis}\}, n_3 \mapsto \{\text{Country}\}, n_4 \mapsto \{\text{Organization}\}$   
 $\lambda_e = r_1 \mapsto \text{birthPlace}, r_2 \mapsto \text{country}, r_3 \mapsto \text{employer}, r_4 \mapsto \text{employer}$   
 $\sigma = (n_1, \text{label}) \mapsto \text{Tim Berners-Lee}, (n_1, \text{birthDate}) \mapsto 1955$   
 $(n_2, \text{label}) \mapsto \text{London}, (n_3, \text{label}) \mapsto \text{UK}, (n_4, \text{label}) \mapsto \text{CERN}$   
 $(r_3, \text{start}) \mapsto 1980, (r_3, \text{end}) \mapsto 1980, (r_4, \text{start}) \mapsto 1984, (r_4, \text{end}) \mapsto 1994$



## Wikibase graphs

Given a mutually disjoint set of items  $\mathcal{Q}$ ,  
 a set of properties  $\mathcal{P}$  and  
 a set of data values  $\mathcal{D}$ ,  
 a *Wikibase graph* is a tuple  $\langle \mathcal{Q}, \mathcal{P}, \mathcal{D}, \rho \rangle$  such that  
 $\rho \subseteq \mathcal{E} \times \mathcal{P} \times \mathcal{V} \times \text{FinSet}(\mathcal{P} \times \mathcal{V})$  where  
 $\mathcal{E} = \mathcal{Q} \cup \mathcal{P}$  is the set of entities which can be subjects of a statement  
 $\mathcal{V} = \mathcal{E} \cup \mathcal{D}$  is the set of possible values of a property.

$\mathcal{Q} = \{ \text{timBl, vintCerf, London, CERN, UK, Spain, PA, Human} \}$   
 $\mathcal{P} = \{ \text{birthDate, birthPlace, country, employer, awarded, start, end, pointTime, togetherWith, instanceOf} \}$   
 $\mathcal{D} = \{ 1984, 1994, 1980, 1955 \}$   
 $\rho = \{ (\text{timBl, instanceOf, Human, {}}), (\text{timBl, birthDate, 1955, {}}), (\text{timBl, birthPlace, London, {}}), (\text{timBl, employer, CERN, { start:1980, end:1980 }}), (\text{timBl, employer, CERN, { start:1984, end:1994 }}), (\text{timBl, awarded, PA, { pointTime: 2002, togetherWith:vintCerf }}), (\text{London, country, UK, {}}), (\text{vintCerf, instanceOf, Human, {}}), (\text{vintCerf, birthPlace, NewHaven, {}}), (\text{CERN, awarded, PA, { pointTime: 2013 }}), (\text{PA, country, Spain, {}}) \}$



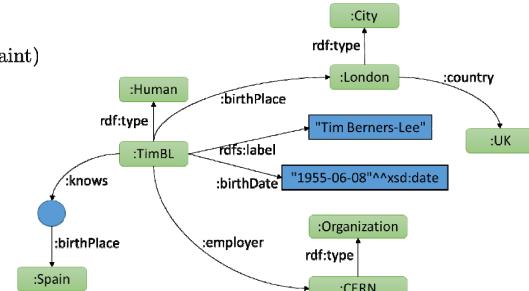
## ShEx for RDF

A ShEx Schema is a tuple  $\langle \mathcal{L}, \delta \rangle$  where

$\mathcal{L}$  set of shape labels

$\delta : \mathcal{L} \rightarrow se$

$se ::= cond$	Basic boolean condition on nodes (node constraint)
$s$	Shape
$se_1 \text{ AND } se_2$	Conjunction
$se_1 \text{ OR } se_2$	Disjunction
$\text{NOT } se$	Negation
$@l$	Shape label reference for $l \in \mathcal{L}$
$s ::= \text{CLOSED } \{te\}$	Closed shape
$\{te\}$	Open shape
$te ::= te_1; te_2$	Each of $te_1$ and $te_2$
$te_1   te_2$	Some of $te_1$ or $te_2$
$te^*$	Zero or more $te$
$\epsilon$	Empty triple expression
$\neg \xrightarrow{p} @l$	Triple constraint with predicate $p$



$\mathcal{L}$	= { Person, Place, Country, Organization, Date }
$\delta(\text{Person})$	= { $\neg \xrightarrow{\text{birthDate}} @Date; \neg \xrightarrow{\text{birthPlace}} @Place;$ $\neg \xrightarrow{\text{employer}} @Organization^*$ }
$\delta(\text{Place})$	= { $\neg \xrightarrow{\text{country}} @Country$ }
$\delta(\text{Country})$	= { }
$\delta(\text{Organization})$	= { }
$\delta(\text{Date})$	= xsd:Date

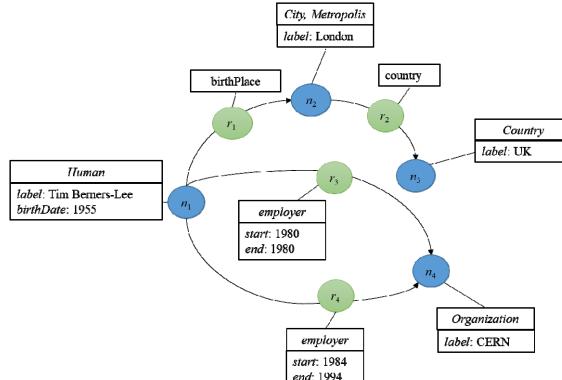
## PShEx for property graphs

A PShEx Schema is a tuple  $\langle \mathcal{L}, \delta \rangle$  where

$\mathcal{L}$  set of shape labels

$\delta : \mathcal{L} \rightarrow se$

$se ::= cond_{t_s}$	Basic boolean condition on set of types $t_s \subseteq \mathcal{T}$
$s$	Shape
$se_1 \text{ AND } se_2$	Conjunction
$se_1 \text{ OR } se_2$	Disjunction
$\text{NOT } se$	Negation
$@l$	Shape label reference for $l \in \mathcal{L}$
$s ::= \text{CLOSED } \{tc\}$	Qualifiers of that node
$\{te\}$	Closed shape
$tc ::= tc_1; tc_2$	Open shape
$te_1   te_2$	Each of $te_1$ and $te_2$
$te^*$	Some of $te_1$ or $te_2$
$\neg \xrightarrow{p} @l, qs$	Zero or more $te$
$qs ::= [ps]$	Triple constraint with property type $p$
$[ps]$	whose nodes satisfy the shape $l$ and qualifiers $qs$
$ps ::= ps_1, ps_2$	Open qualifier specifiers $ps$
$ps_1   ps_2$	Closed qualifier specifiers $ps$
$ps^*$	Each of $ps_1$ and $ps_2$
$p : cond_v$	OneOf of $ps_1$ or $ps_2$
$cond_v$	zero or more $ps$
$cond_{v_s}$	Property $p$ with value conforming to $cond_v$
	$cond_{v_s}$ is a boolean condition on sets of values $v_s \subseteq \mathcal{V}$



$\mathcal{L}$	= { Person, Place, Country, Org }
$\delta(\text{Person})$	= { hasTypeHuman AND [label : String, birthDate : Date] AND { $\neg \xrightarrow{\text{birthPlace}} @Place;$ $\neg \xrightarrow{\text{employer}} @Org [start : Date, end : Date]^*$ } }
$\delta(\text{Place})$	= { [label : String] AND { $\neg \xrightarrow{\text{country}} @Country$ } }
$\delta(\text{Country})$	= { hasTypeCountry AND [label : String] {} }
$\delta(\text{Org})$	= { hasTypeOrganization AND [label : String] {} }

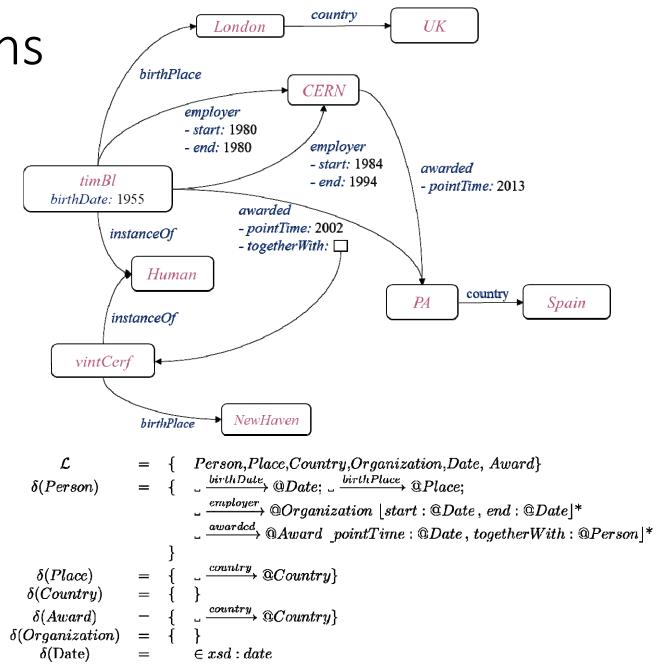
## WShEx for Wikibase graphs

A WShEx Schema is a tuple  $(\mathcal{L}, \delta)$  where

$\mathcal{L}$  set of shape labels

$\delta : \mathcal{L} \rightarrow se$

$se ::=$	cond	Basic boolean condition on nodes (node constraint)
	$s$	Shape
	$se_1 \text{ AND } se_2$	Conjunction
	$se_1 \text{ OR } se_2$	Disjunction
	$\text{NOT } se$	Negation
	$@l$	Shape label reference for $l \in \mathcal{L}$
$s ::=$	$\text{CLOSED } s'$	Closed shape
	$s'$	Open shape
$s' ::=$	{ tc }	Shape definition
$te ::=$	$te_1; te_2$	Each of $te_1$ and $te_2$
	$te_1   te_2$	Some of $te_1$ or $te_2$
	$te^*$	Zero or more $te$
	$\exists p @l qs$	Triple constraint with predicate $p$ value conforming to $l$ and qualifier specifier $qs$
$qs ::=$	$[ps]$	Empty triple expression
	$[ps]$	Open property specifier
	$[ps]$	Closed property specifier
$ps ::=$	$ps, ps$	$EachOf$ property specifiers
	$ps   ps$	$OneOf$ property specifiers
	$ps^*$	zero or more property specifiers
	$\epsilon$	Empty property specifier
	$p:@l$	Property $p$ with value conforming to shape $l$

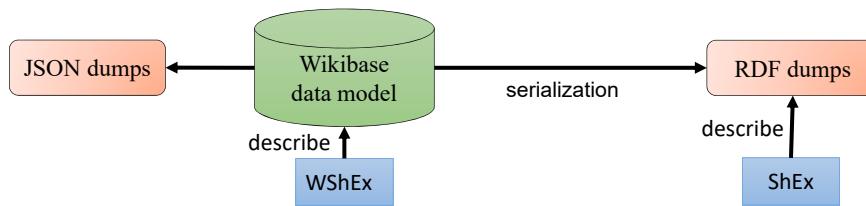


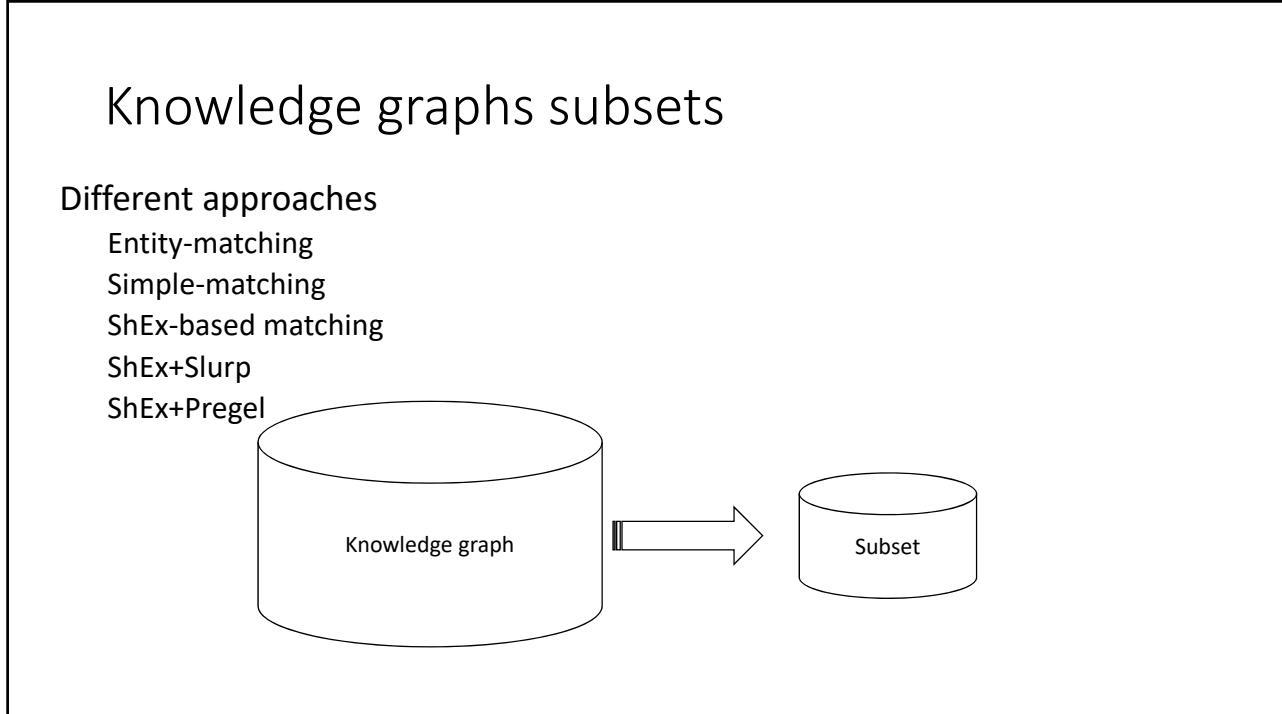
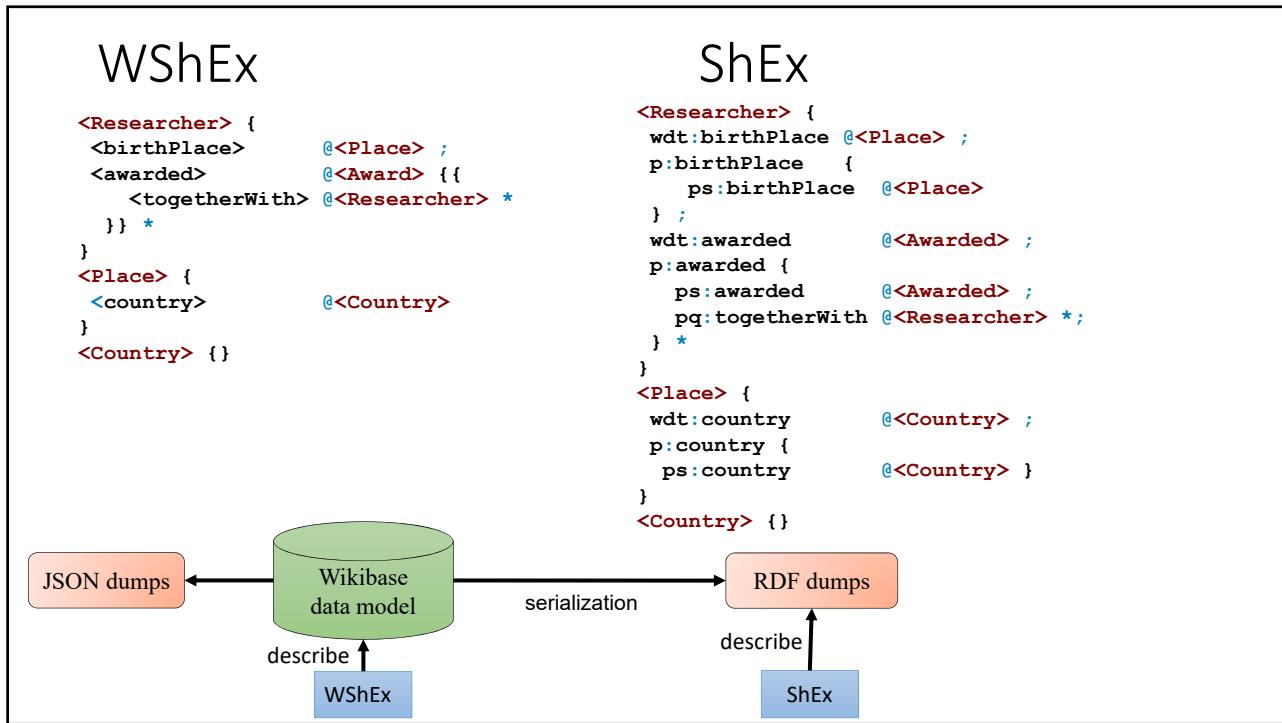
## What's the role of WShEx

WShEx schemas describe Wikibase data model

Closer to JSON dumps

ShEx (entity schemas) describe RDF serialization of Wikibase data model





## Entity-matching

Directly matches an entity like an ítem, property, etc.

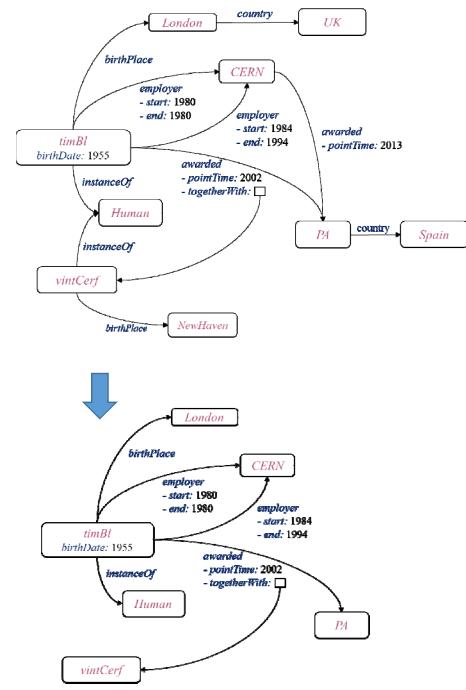
Example:  $Q_s = \{timBl\}$

$$Q' = \{timBl, CERN, vintCerf, PA\}$$

$$\mathcal{P}' = \{birthDate, birthPlace, employer, awarded, start, end, togetherWith\}$$

$$\mathcal{D}' = \{1984, 1994, 1980, 1955\}$$

$$\begin{aligned} S' = & \{(timBl, birthDate, 1955, \{\}), \\ & (timBl, birthPlace, London, \{\}), \\ & (timBl, employer, CERN, \{start : 1980, end : 1980\}), \\ & (timBl, employer, CERN, \{start : 1984, end : 1994\}), \\ & (timBl, awarded, PA, \{togetherWith : vintCerf\}), \\ & (vintCerf, awarded, PA, \{togetherWith : timBl\})\} \end{aligned}$$



## Simple matching

Defines a matching language:

$$\begin{aligned} m ::= & \text{subject}(e) && \text{Subject } e \in \mathcal{E} \\ & \mid \text{property}(p) && \text{Property } p \in \mathcal{P} \\ & \mid \text{value}(v) && \text{Value } v \in \mathcal{V} \\ & \mid \text{qualifier}(p, v) && \text{Qualifier with property } p \in \mathcal{P} \text{ and value } v \in \mathcal{V} \\ & \mid \text{qualifiedProp}(p) && \text{Qualifier with property } p \in \mathcal{P} \\ & \mid \text{qualifiedValue}(v) && \text{Qualifier with value } v \in \mathcal{V} \end{aligned}$$

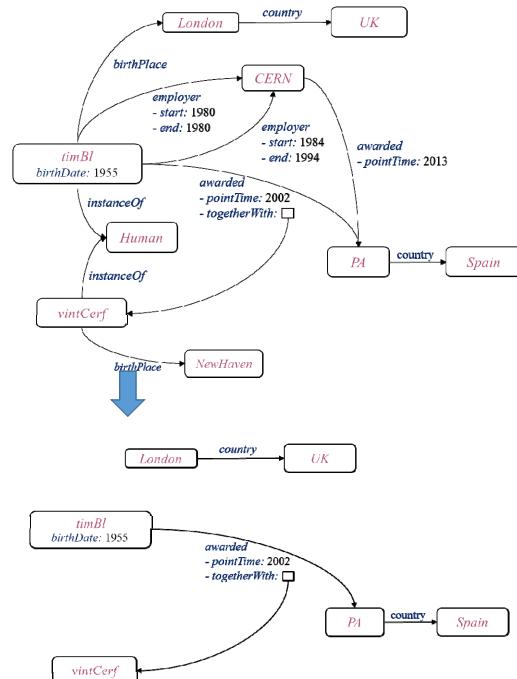
Example:  $M_s = \{\text{property(country)}, \text{qualifiedProp(togetherWith)}\}$

$$Q' = \{PA, Spain, London, UK, timBl, vintCerf\}$$

$$\mathcal{P}' = \{country, awarded, togetherWith\}$$

$$\mathcal{D}' = \{\}$$

$$\begin{aligned} S' = & \{(timBl, awarded, PA, \{togetherWith : vintCerf\}), \\ & (PA, country, Spain, \{\}), \\ & (London, country, UK, \{\})\} \end{aligned}$$



## ShEx-based matching

Uses ShEx definitions to match entities

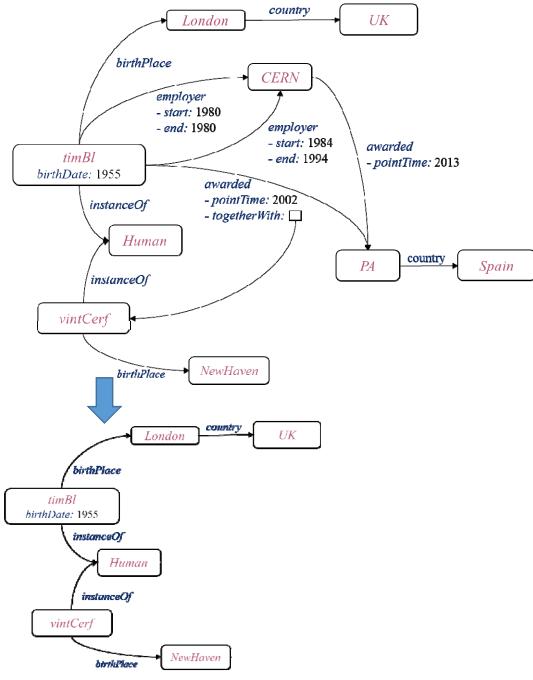
Replaces shape references by *true*

Doesn't require graph traversal

```

 $\mathcal{L} = \{ Researcher, Place, Country, Date, Human \}$ 
 $\delta(Researcher) = \{ \text{instanceOf} @Human;$ 
 $\quad \text{birthDate} @Date?;$ 
 $\quad \text{birthPlace} @Place$ 
 $\}$ 
 $\delta(Place) = \{ \text{country} @Country \}$ 
 $\delta(Date) = \{ \in xsd:date \}$ 
 $\delta(Human) = \{ \text{Human} \}$ 
 $\mathcal{S} = \{ (timBl, instanceOf, Human), \{ \},$ 
 $\quad (timBl, birthDate, 1955), \{ \},$ 
 $\quad (timBl, birthPlace, London), \{ \},$ 
 $\quad (London, country, UK), \{ \},$ 
 $\quad (vintCerf, instanceOf, Human), \{ \}$ 
 $\quad (vintCerf, birthPlace, NewHaven), \{ \}$ 
 $\}$ 

```



## ShEx-based matching

Can be implemented replacing shape references by *true*

```

 $\delta(Researcher) = \{ \text{instanceOf} @Human;$ 
 $\quad \text{birthDate} @Date?;$ 
 $\quad \text{birthPlace} @Place$ 
 $\}$ 
 $\delta(Place) = \{ \text{country} @Country \}$ 
 $\delta(Date) = \{ \in xsd:date \}$ 
 $\delta(Human) = \{ \text{Human} \}$ 

```

```

 $\delta(Researcher) = \{ \text{instanceOf} \rightarrow \text{true};$ 
 $\quad \text{birthDate} \rightarrow \text{true?};$ 
 $\quad \text{birthPlace} \rightarrow \text{true}$ 
 $\}$ 
 $\delta(Place) = \{ \text{country} \rightarrow \text{true} \}$ 

```

ShEx-based matching has been implemented in WDSub

It allows sequential processing of the dump

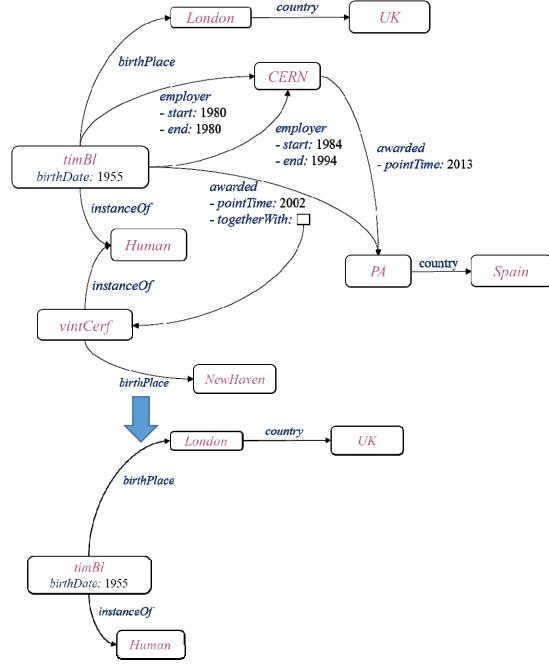
It can process the latest Wikidata dump in 5h 15'

## ShEx+slurp

Slurp = Shex option that collects nodes/triples while validating

Example:

$$\begin{aligned}
 \mathcal{L} &= \{ \text{Researcher}, \text{Place}, \text{Country}, \text{Date}, \text{Human} \} \\
 \delta(\text{Researcher}) &= \{ \text{instanceOf} \rightarrow @\text{Human}; \\
 &\quad \text{birthDate} \rightarrow @\text{Date}; \\
 &\quad \text{birthPlace} \rightarrow @\text{Place} \\
 &\} \\
 \delta(\text{Place}) &= \{ \text{country} \rightarrow @\text{Country} \} \\
 \delta(\text{Country}) &= \{ \} \\
 \delta(\text{Date}) &= \in \text{xsd: date} \\
 \delta(\text{Human}) &= \in \{\text{Human}\}
 \end{aligned}$$



## ShEx+Slurp

Formal definition generalizes ShEx semantics

New conformance relation:

$$\mathcal{G}, n, \tau \models se \rightsquigarrow \mathcal{G}'$$

$$\begin{aligned}
 \text{EachOf} &\frac{(ts_1, ts_2) \in \text{part}(ts) \quad \mathcal{G}, ts_1, \tau \Vdash te_1 \rightsquigarrow \mathcal{G}_1 \quad \mathcal{G}, ts_2, \tau \Vdash te_2 \rightsquigarrow \mathcal{G}_2}{\mathcal{G}, ts, \tau \Vdash te_1 | te_2 \rightsquigarrow \mathcal{G}_1 \cup \mathcal{G}_2} \\
 \text{OneOf}_1 &\frac{\mathcal{G}, ts, \tau \Vdash te_1 \rightsquigarrow \mathcal{G}_1}{\mathcal{G}, ts, \tau \Vdash te_1 | te_2 \rightsquigarrow \mathcal{G}_1} \quad \text{OneOf}_2 \frac{\mathcal{G}, ts, \tau \Vdash te_2 \rightsquigarrow \mathcal{G}_2}{\mathcal{G}, ts, \tau \Vdash te_1 | te_2 \rightsquigarrow \mathcal{G}_2} \\
 \text{Star}_1 &\frac{}{\mathcal{G}, \emptyset, \tau \Vdash te^* \rightsquigarrow \emptyset} \\
 \text{Star}_2 &\frac{(ts_1, ts_2) \in \text{part}(ts) \quad \mathcal{G}, ts_1, \tau \Vdash te \rightsquigarrow \mathcal{G}_1 \quad \mathcal{G}, ts_2, \tau \Vdash te^* \rightsquigarrow \mathcal{G}_2}{\mathcal{G}, ts, \tau \Vdash te^* \rightsquigarrow \mathcal{G}_1 \cup \mathcal{G}_2} \\
 \text{TripleConstraint} &\frac{ts = \{(x, p, y, s)\} \quad \mathcal{G}, y, \tau \models @l \rightsquigarrow (\mathcal{V}, \mathcal{E}) \quad \mathcal{G}, s, \tau \Vdash qs \rightsquigarrow (qs', G_{qs})}{\mathcal{G}, ts, \tau \Vdash \xrightarrow{P} @l \, qs \rightsquigarrow (\mathcal{V} \cup \{x\} \cup \{y\}, \mathcal{E} \cup (x, p, y, qs')) \cup G_{qs}}
 \end{aligned}$$

$$\begin{array}{c}
 \text{Cond} \frac{\text{cond}(n) = \text{true}}{\mathcal{G}, n, \tau \models \text{cond} \rightsquigarrow \{\{n\}, \{\}\}} \quad \text{AND} \frac{\mathcal{G}, n, \tau \models se_1 \rightsquigarrow \mathcal{G}_1 \quad \mathcal{G}, n, \tau \models se_2 \rightsquigarrow \mathcal{G}_2}{\mathcal{G}, n, \tau \models se_1 \text{ AND } se_2 \rightsquigarrow \mathcal{G}_1 \cup \mathcal{G}_2} \\
 \text{ClosedShape} \frac{\text{neighs}(n, \mathcal{G}) = ts \quad \mathcal{G}, ts, \tau \Vdash s' \rightsquigarrow \mathcal{G}'}{\mathcal{G}, n, \tau \Vdash \text{CLOSED } s' \rightsquigarrow \mathcal{G}'} \\
 \text{OpenShape} \frac{\text{ts} = \{(x, p, y) \in \text{neighs}(n, \mathcal{G}) \mid p \in \text{preds}(te)\} \quad \mathcal{G}, ts, \tau \Vdash s' \rightsquigarrow \mathcal{G}'}{\mathcal{G}, n, \tau \models s' \rightsquigarrow \mathcal{G}'} \\
 \text{OpenQs} \frac{s' = \{(p, v) \in s \mid p \in \text{preds}(ps)\} \quad \mathcal{G}, s', \tau \Vdash ps \rightsquigarrow (qs, \mathcal{G}')}{\mathcal{G}, s, \tau \Vdash [ps] \rightsquigarrow (qs, \mathcal{G}')}, \quad \text{CloseQs} \frac{\mathcal{G}, s, \tau \Vdash ps \rightsquigarrow (qs, \mathcal{G}')}{\mathcal{G}, s, \tau \Vdash [ps] \rightsquigarrow (qs, \mathcal{G}')} \\
 \text{EachOfQs} \frac{\mathcal{G}, s, \tau \Vdash ps_1 \rightsquigarrow (qs_1, \mathcal{G}_1) \quad \mathcal{G}, s, \tau \Vdash ps_2 \rightsquigarrow (qs_2, \mathcal{G}_2)}{\mathcal{G}, s, \tau \Vdash ps_1 | ps_2 \rightsquigarrow (qs_1 \cup qs_2, \mathcal{G}_1 \cup \mathcal{G}_2)} \\
 \text{OneOfQs}_1 \frac{\mathcal{G}, s, \tau \Vdash ps_1 \rightsquigarrow (qs_1, \mathcal{G}_1)}{\mathcal{G}, s, \tau \Vdash ps_1 | ps_2 \rightsquigarrow (qs_1, \mathcal{G}_1)}, \quad \text{OneOfQs}_2 \frac{\mathcal{G}, s, \tau \Vdash ps_2 \rightsquigarrow (qs_2, \mathcal{G}_2)}{\mathcal{G}, s, \tau \Vdash ps_1 | ps_2 \rightsquigarrow (qs_2, \mathcal{G}_2)} \\
 \text{StarQs}_1 \frac{}{\mathcal{G}, \emptyset, \tau \Vdash ps^* \rightsquigarrow (\{\}, \emptyset)} \\
 \text{StarQs}_2 \frac{(s_1, s_2) \in \text{part}(s) \quad \mathcal{G}, s_1, \tau \Vdash ps \rightsquigarrow (qs_1, \mathcal{G}_1) \quad \mathcal{G}, s_2, \tau \Vdash ps^* \rightsquigarrow (qs_2, \mathcal{G}_2)}{\mathcal{G}, s, \tau \Vdash ps^* \rightsquigarrow (qs_1 \cup qs_2, \mathcal{G}_1 \cup \mathcal{G}_2)} \\
 \text{EmptyQs} \frac{}{\mathcal{G}, \emptyset, \tau \Vdash \epsilon \rightsquigarrow (\{\}, \emptyset)}, \quad \text{PropertyQs} \frac{s = \{(p, v)\} \quad \mathcal{G}, v, \tau \models @l \rightsquigarrow \mathcal{G}'}{\mathcal{G}, s, \tau \Vdash p : @l \rightsquigarrow (\{p : v\}, \mathcal{G}'')}
 \end{array}$$

## ShEx+Slurp

Implemented by shex.js and pyshex

Only for ShEx, not for PShEx or WShEx

It does graph traversal while validating

Difficult to use for processing Wikidata dumps

Subsets include only valid data according to the ShEx

Problems:

Difficult to run behind SPARQL endpoint

Naïve implementation generates too many requests

Optimizations to improve number of SPARQL queries can generate timeouts

## ShEx+Pregel

Large scale validation

Based on Pregel algorithm

Developed by Google 2010

Based on Bulk Synchronous Parallel model

Supersteps = units of parallel computation

Communication through messages

Implemented using Apache Spark GraphX library

## Think like a vertex

Vertex-centric approach

Each vertex has an id and a state

All vertexes receive an initial message

While messages received do

- Run vProg to update vertex state according to messages received

- Run sendMsg for each triplet to generate messages

- Merge messages by Vertex

## GraphX

Graph API from Apache Spark

RDD (Resilient distributed datasets)

MapReduce in memory

$\text{Graph}[\mathcal{V}, \mathcal{E}]$  = graph with vertices  $\mathcal{V}$  and edges  $\mathcal{E}$

- Vertices are represented as  $\text{RDD}[(Id, \mathcal{V})]$  where  $Id = \text{Long}$
- Edges are represented as  $\text{RDD}[(Id, Id, \mathcal{E})]$
- triplets* view represents edges as collections  $\text{RDD}[(\mathcal{V}, \mathcal{E}, \mathcal{V})]$ .

Several built-in operators:

- `mapVertices(g: Graph[V,E], f: (Id,V) → V): Graph[V,E]`
- `mapReduceTriples(g:Graph[V,E], m: (V,E,V) → [(Id,M)], r:(M,M) → M):RDD[(Id,M)]`
- `joinVertices(g:Graph[V,E], msgs:RDD[(Id, M)], f:(Id, V,M) → V): Graph[V,E]`
- pregel (see next slide)
- ...

## GraphX Pregel pseudo-code

---

**Algorithm 1:** Pregel algorithm pseudocode as implemented in GraphX

---

**Input parameters:**

```

g: Graph[V,E]
initialMsg: M
vProg: (Id,V,M)→V
sendMsg: Triplet→[(Id,M)]
mergeMsg: (M,M)→M

```

**Output:** g:Graph[V,E]

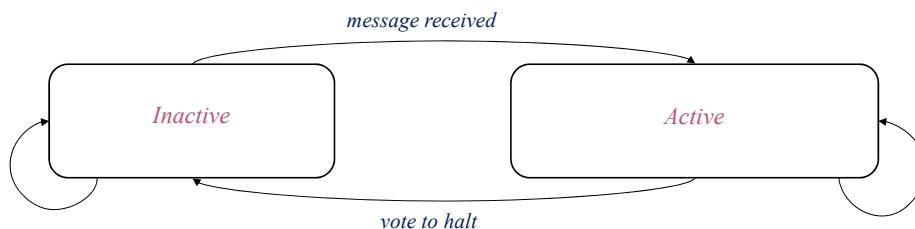
```

1 g = mapVertices(g,λ(id,v)→vProg(id,v,initialMsg))
2 msgs = mapReduceTriples(g,sendMsg,mergeMsg)
3 while size(msgs)> 0 do
4   g = joinVertices(g,msgs,vProg)
5   msgs = mapReduceTriples(g,sendMsg,mergeMsg)
6 return g

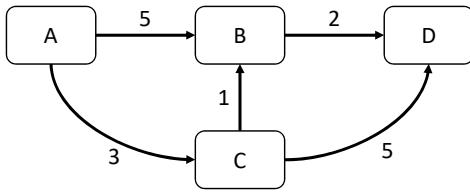
```

---

## Pregel's vertex state diagram



## Shortest path to a node



```

val initialGraph = graph.mapVertices(
  (id, _) => if (id == source) 0 else ∞
)

initialMsg = ∞

def vProg(id: VertexId, dist: Int, msg: Int) = min(dist, msg)

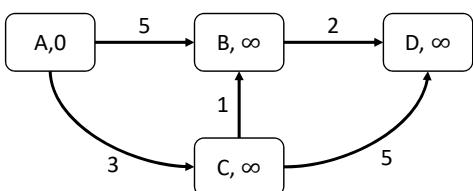
def sendMsg(t: Triplet) = {
  if (t.srcAttr + t.attr < t.dstAttr) {
    (t.dstId, t.srcAttr + t.attr)
  } else { }
}

def mergeMsg(a: Int, b: Int) = min(a, b)

initialGraph.pregel(initialMsg)
  (vProg, sendMsg, mergeMsg)
  
```

## Shortest path to a node

initialGraph



```

val initialGraph = graph.mapVertices(
  (id, _) => if (id == source) 0 else ∞
)

initialMsg = ∞

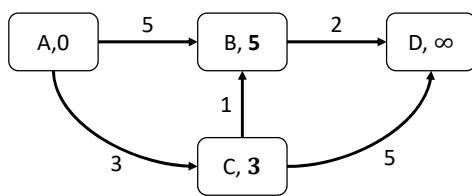
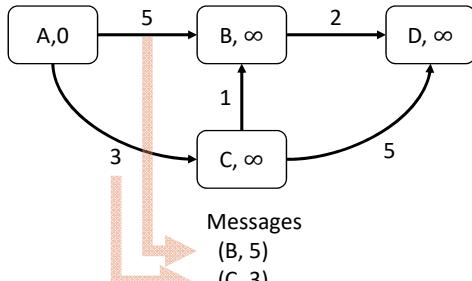
def vProg(id: VertexId, dist: Int, msg: Int) = min(dist, msg)

def sendMsg(t: Triplet) = {
  if (t.srcAttr + t.attr < t.dstAttr) {
    (t.dstId, t.srcAttr + t.attr)
  } else { }
}

def mergeMsg(a: Int, b: Int) = min(a, b)

initialGraph.pregel(initialMsg)
  (vProg, sendMsg, mergeMsg)
  
```

## Shortest path to a node



```

val initialGraph = graph.mapVertices(
  (id, _) => if (id == source) 0 else ∞
)

initialMsg = ∞

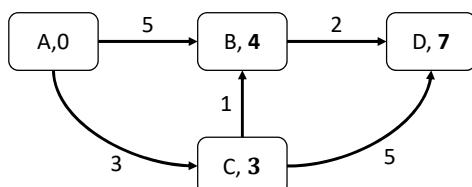
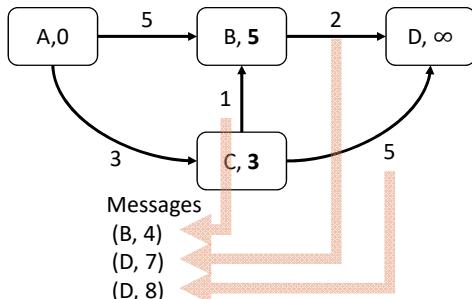
def vProg(id: VertexId, dist: Int, msg: Int) =
  min(dist, msg)

def sendMsg(t: Triplet) = {
  if (t.srcAttr + t.attr < t.dstAttr) {
    (t.dstId, t.srcAttr + t.attr)
  } else { }
}

def mergeMsg(a: Int, b: Int) = min(a, b)

initialGraph.pregel(initialMsg)
  (vProg, sendMsg, mergeMsg)
  
```

## Shortest path to a node



```

val initialGraph = graph.mapVertices(
  (id, _) => if (id == source) 0 else ∞
)

initialMsg = ∞

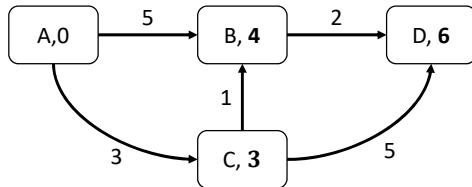
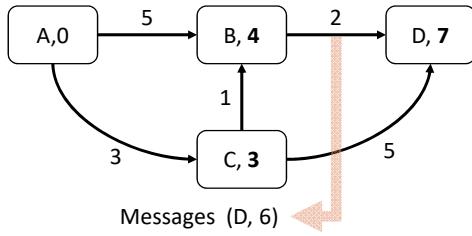
def vProg(id: VertexId, dist: Int, msg: Int) =
  min(dist, msg)

def sendMsg(t: Triplet) = {
  if (t.srcAttr + t.attr < t.dstAttr) {
    (t.dstId, t.srcAttr + t.attr)
  } else { }
}

def mergeMsg(a: Int, b: Int) = min(a, b)

initialGraph.pregel(initialMsg)
  (vProg, sendMsg, mergeMsg)
  
```

## Shortest path to a node



```

val initialGraph = graph.mapVertices(
  (id, _) => if (id == source) 0 else ∞
)

initialMsg = ∞

def vProg(id: VertexId, dist: Int, msg: Int) =
  min(dist, msg)

def sendMsg(t: Triplet) = {
  if (t.srcAttr + t.attr < t.dstAttr) {
    (t.dstId, t.srcAttr + t.attr)
  } else { }
}

def mergeMsg(a: Int, b: Int) = min(a, b)

initialGraph.pregel(initialMsg)
  (vProg, sendMsg, mergeMsg)
  
```

## Idea: Validate graphs using Pregel?

Associate each node with a status map

Status map contains information about node conformance to shapes

Messages =

Requests to validate

Request to wait for neighbours

Information about validated/failed neighbours

Initial message = request to validate start shape

In each iteration we collect information from neighbours conformance

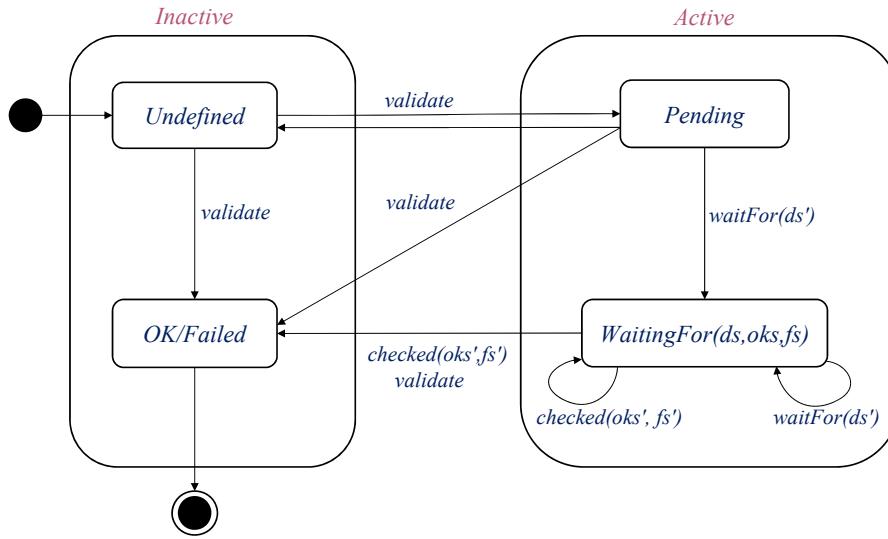
We provide 3 parameters

checkLocal: tries to check conformance of a node locally or return pending neighbours

checkNeighs: checks neighbours with regular expression defined by a shape

tripleConstraints: return the triple constraints associated with a shape label

## State diagram



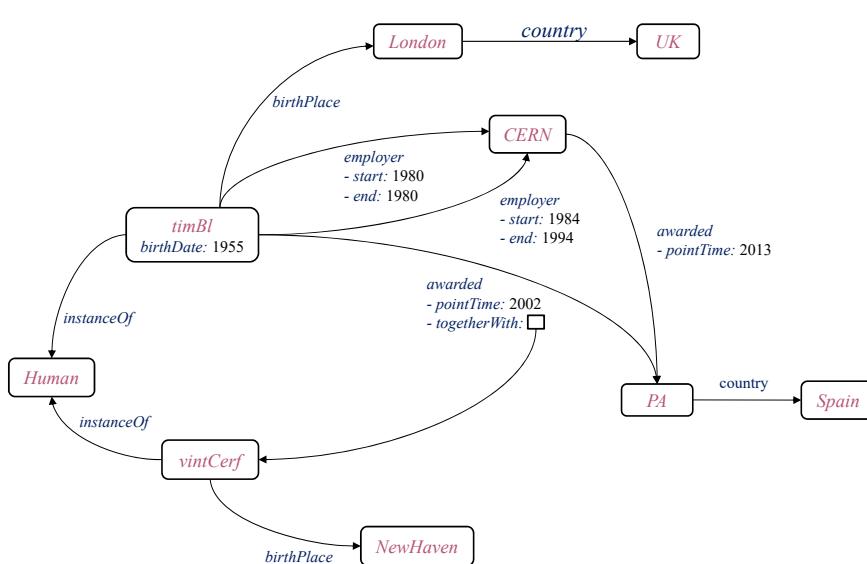
## Trace of PSchema

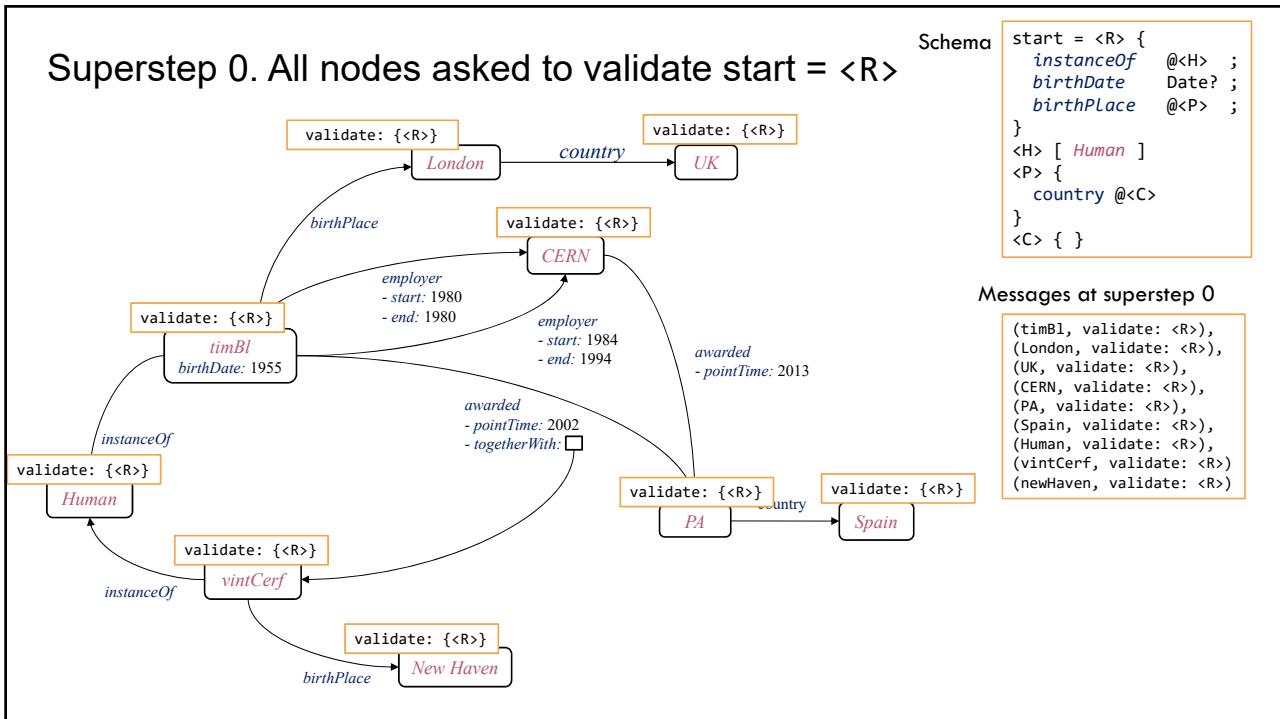
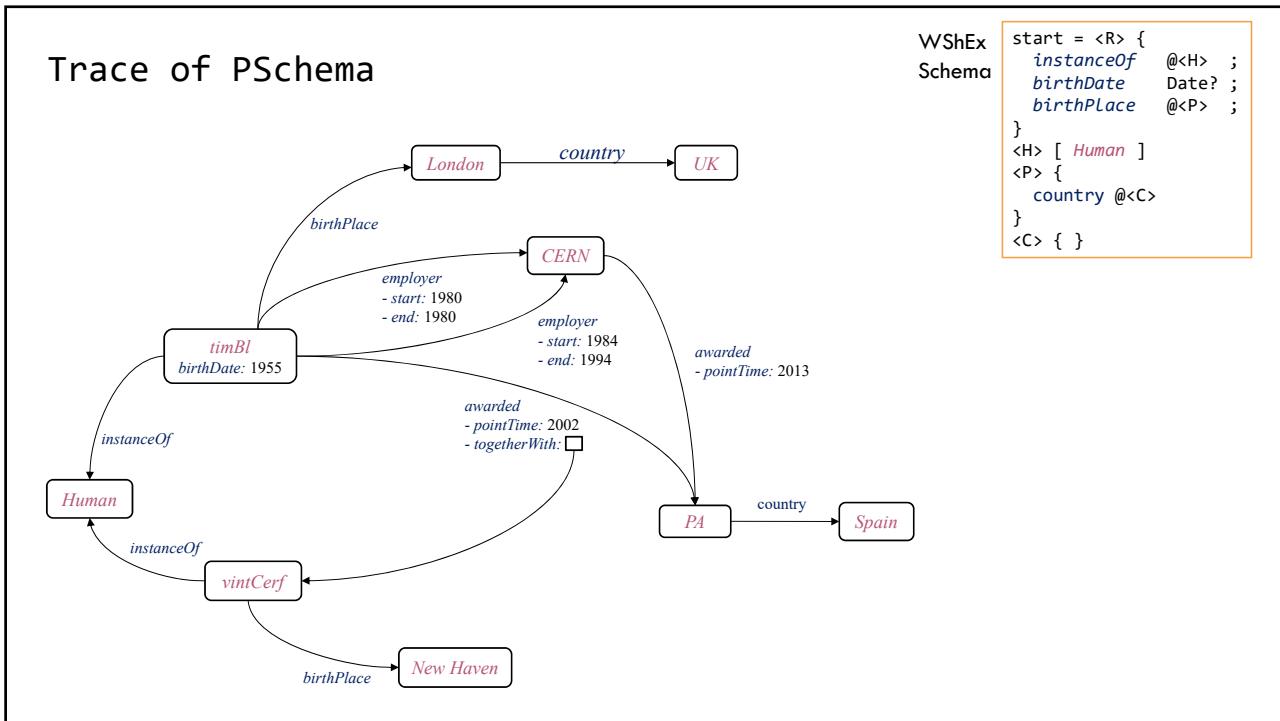
WShEx Schema

```

start = <Researcher> {
  instanceOf @<Human> ;
  birthDate Date? ;
  birthPlace @<Place> ;
}
<Human> [ Human ]
<Place> {
  country @<Country>
}
<Country> { }

```





## Superstep 0. After vprog

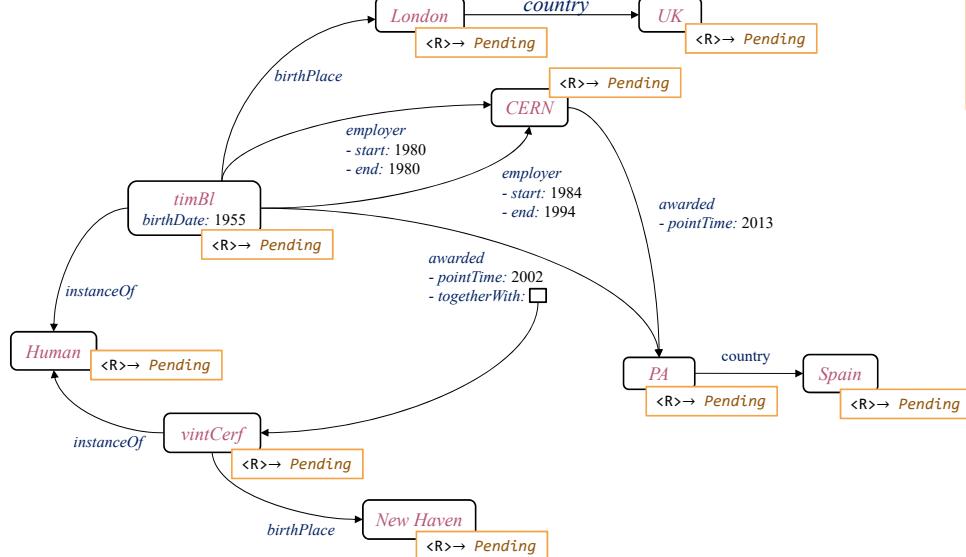
Schema

```

start = <R> {
  instanceOf @<H> ;
  birthDate Date? ;
  birthPlace @<P> ;
}

<H> [ Human ]
<P> {
  country @<C>
}
<C> { }

```



## Superstep 1. Collecting messages

Schema

```

start = <R> {
  instanceOf @<H> ;
  birthDate Date? ;
  birthPlace @<P> ;
}

<H> [ Human ]
<P> {
  country @<C>
}
<C> { }

```

Msgs

```

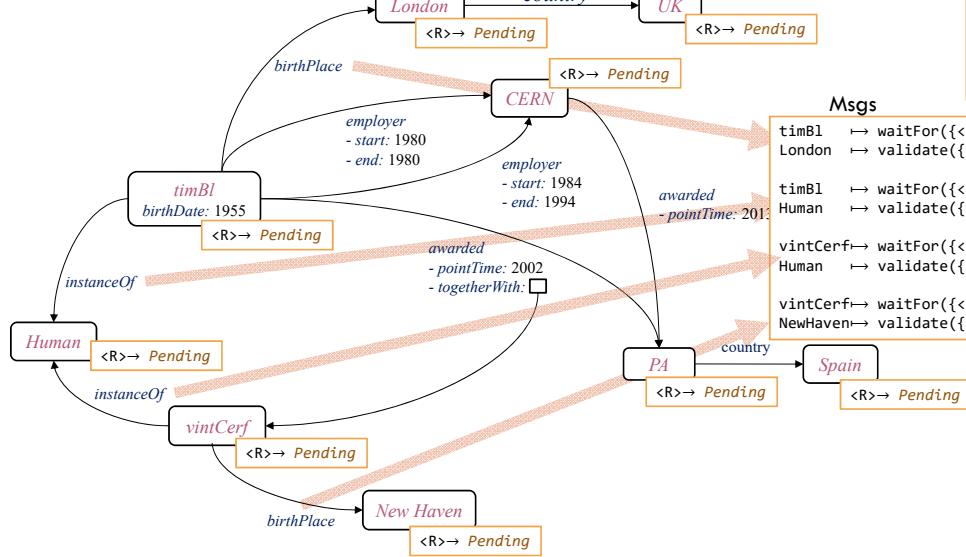
timBL → waitFor({<R>, (London, birthPlace, <P>)})
London → validate({<P>})

timBL → waitFor({<R>, (Human, instanceOf, <H>)})
Human → validate({<H>})

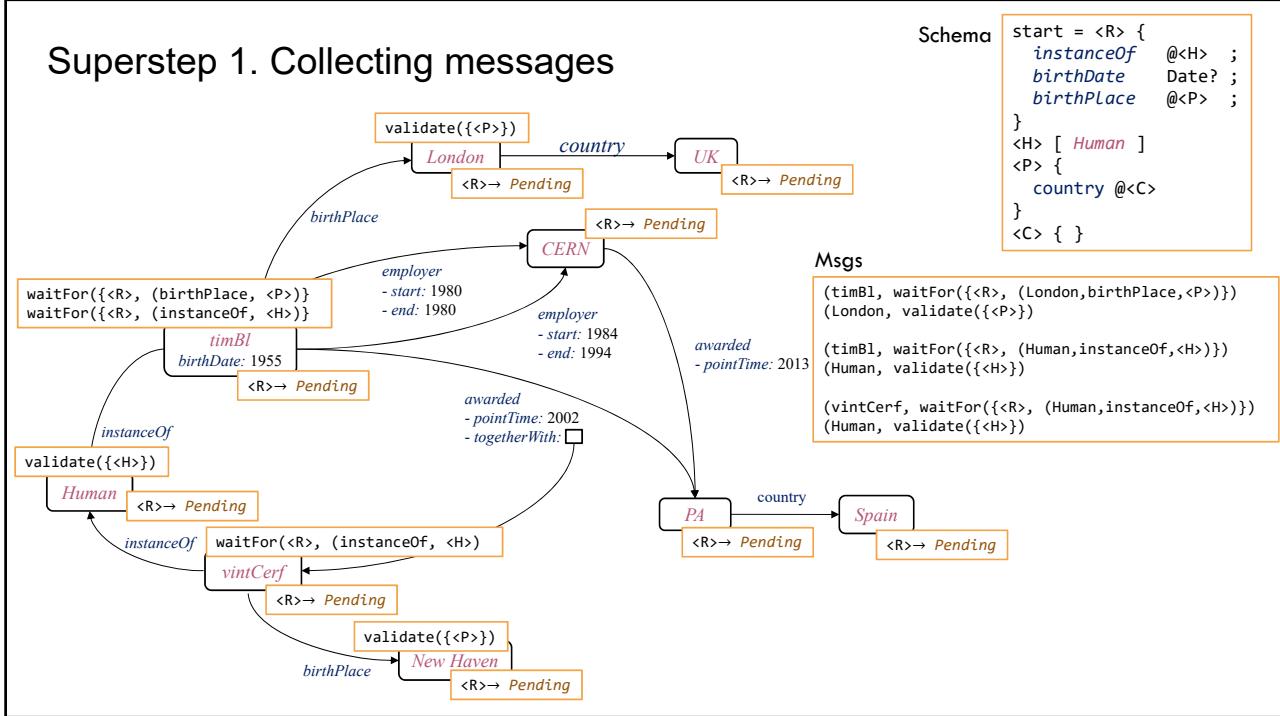
vintCerf → waitFor({<R>, (Human, instanceOf, <H>)})
Human → validate({<H>})

vintCerf → waitFor({<R>, (NewHaven, instanceOf, <H>)})
NewHaven → validate({<P>})

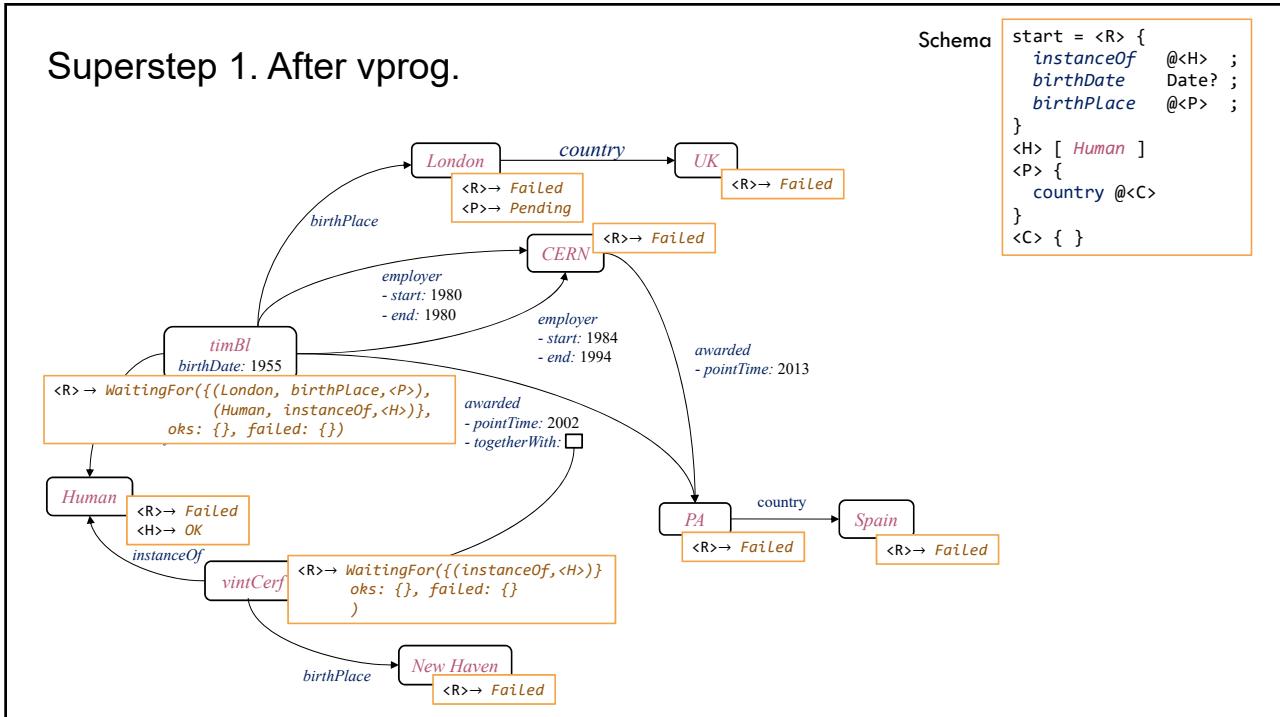
```



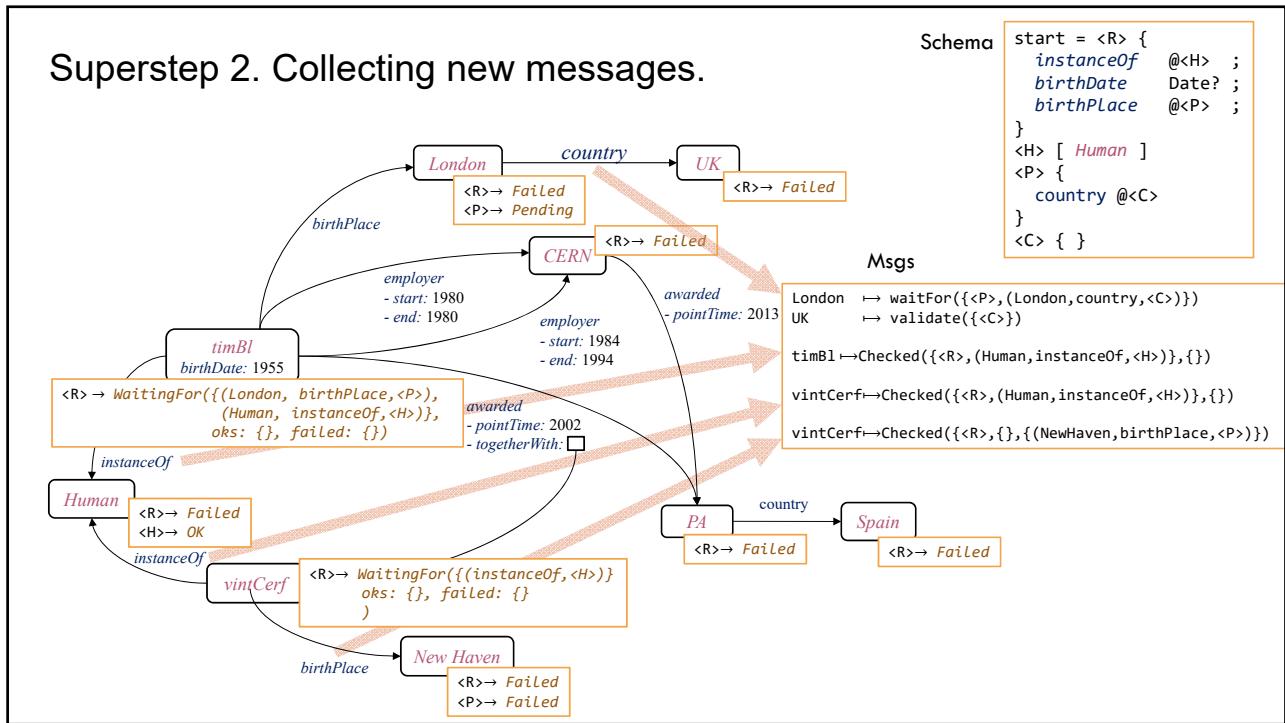
## Superstep 1. Collecting messages



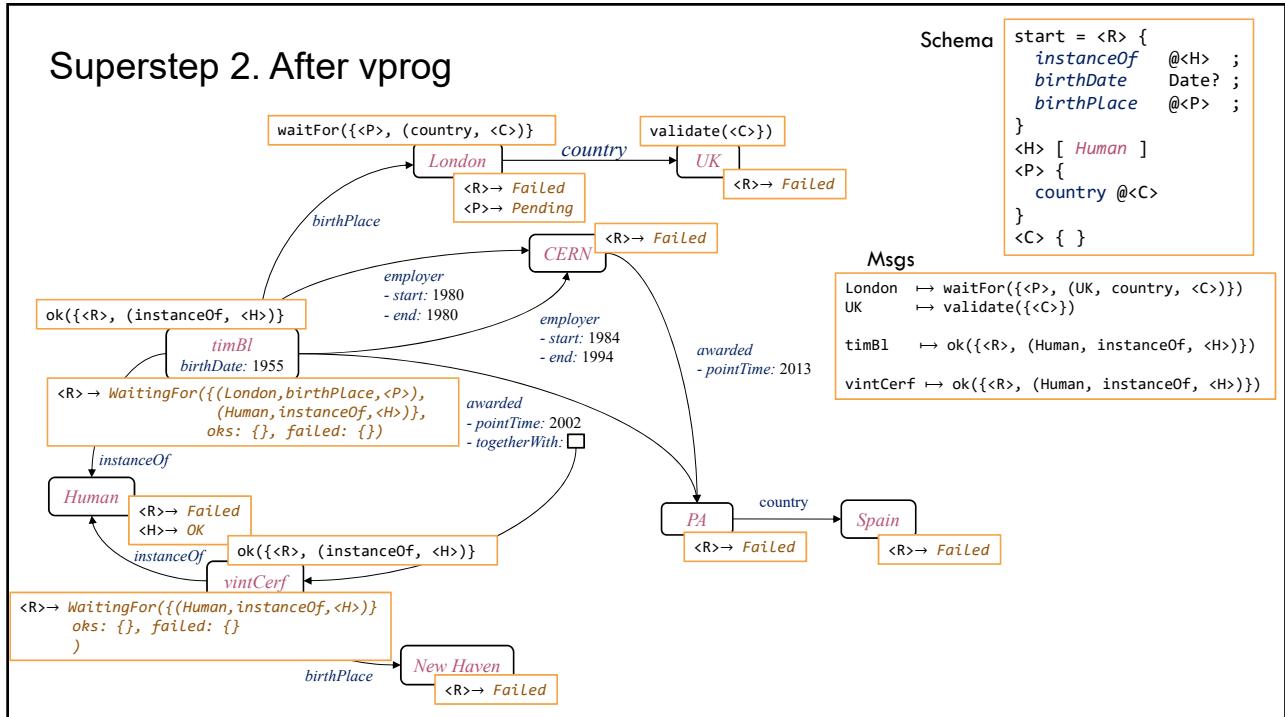
## Superstep 1. After vprog.



## Superstep 2. Collecting new messages.



## Superstep 2. After vprog



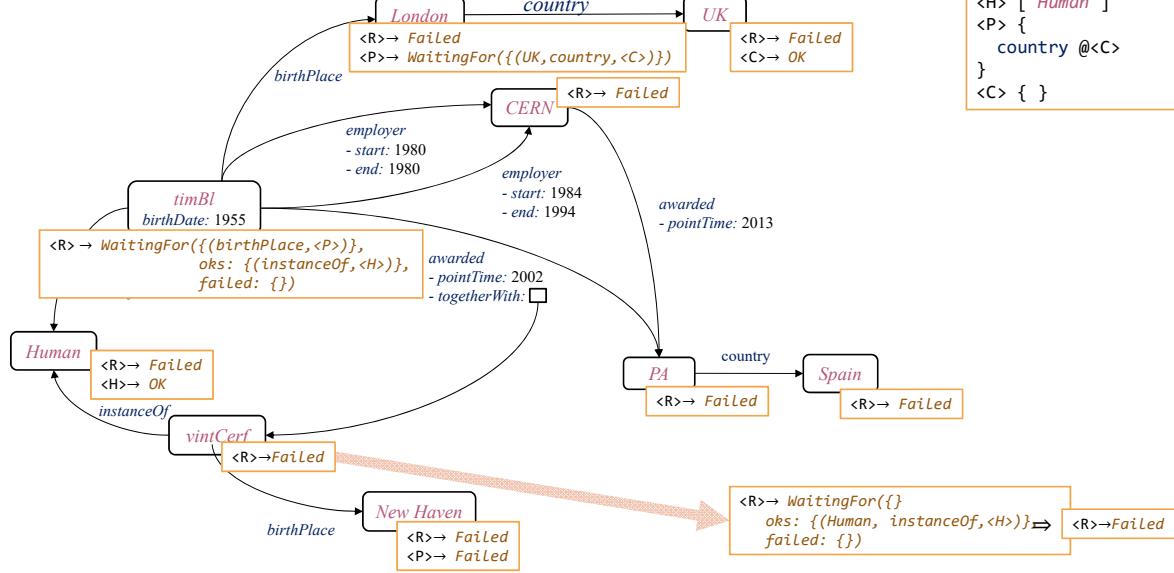
## Superstep 2. After vprog

Schema

```

start = <R> {
    instanceOf @<H> ;
    birthDate Date? ;
    birthPlace @<P> ;
}
<H> [ Human ]
<P> {
    country @<C>
}
<C> { }

```



## Superstep 3. Collecting messages

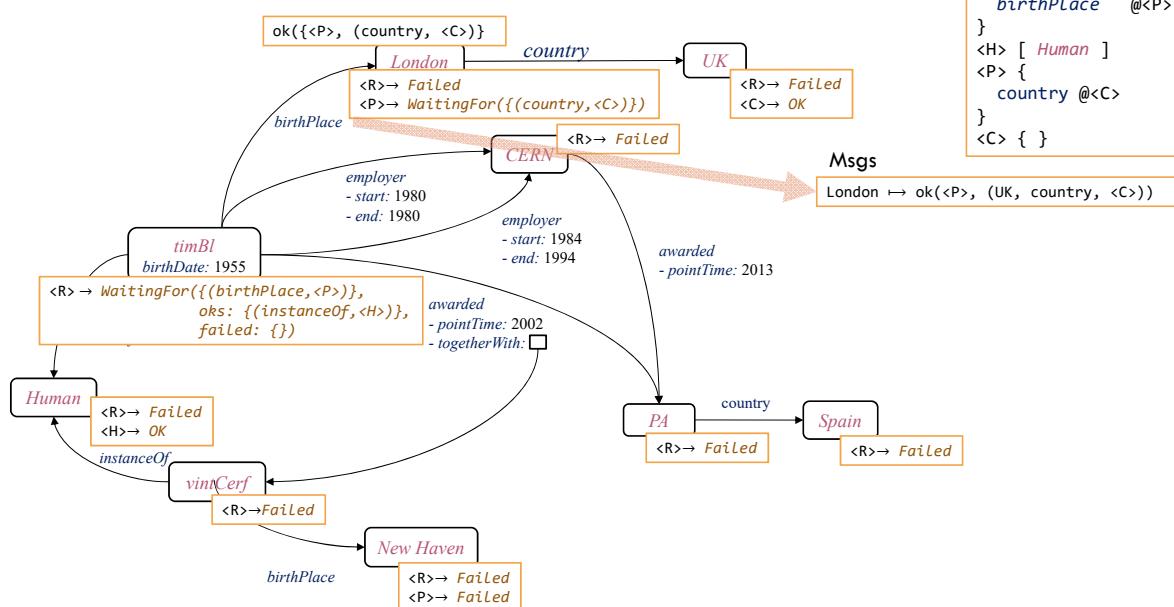
Schema

```

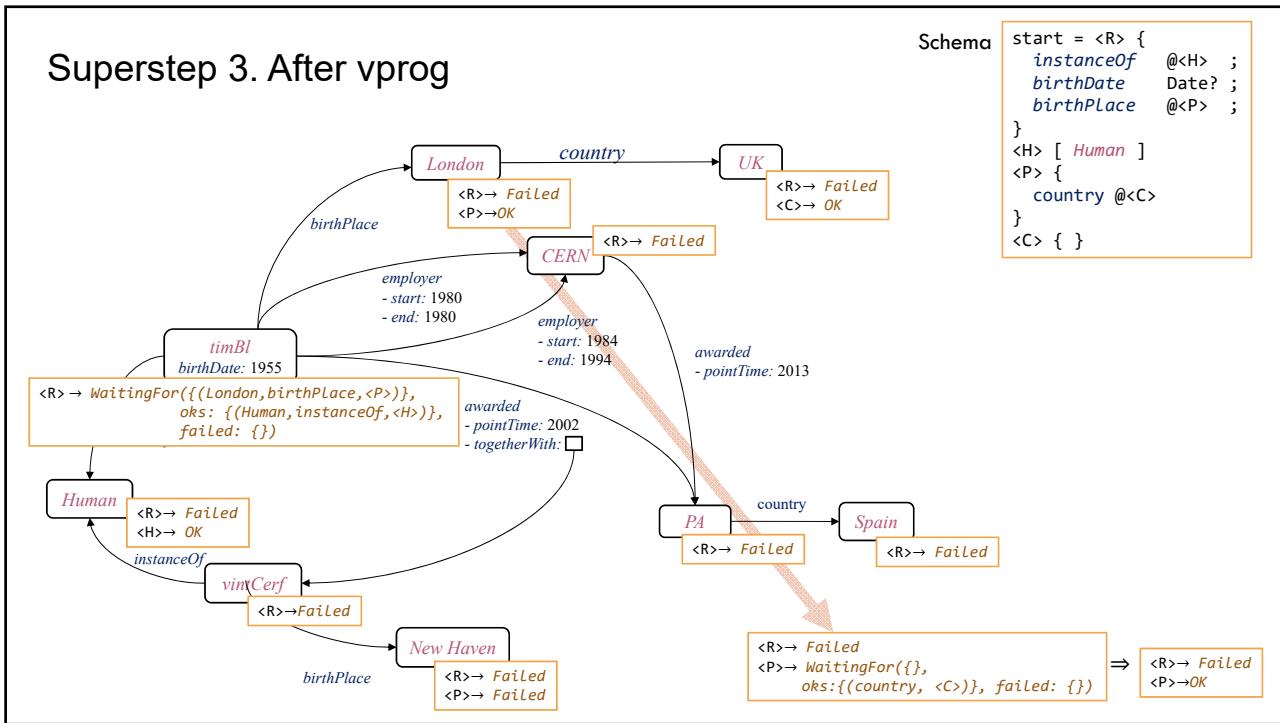
start = <R> {
    instanceOf @<H> ;
    birthDate Date? ;
    birthPlace @<P> ;
}
<H> [ Human ]
<P> {
    country @<C>
}
<C> { }

```

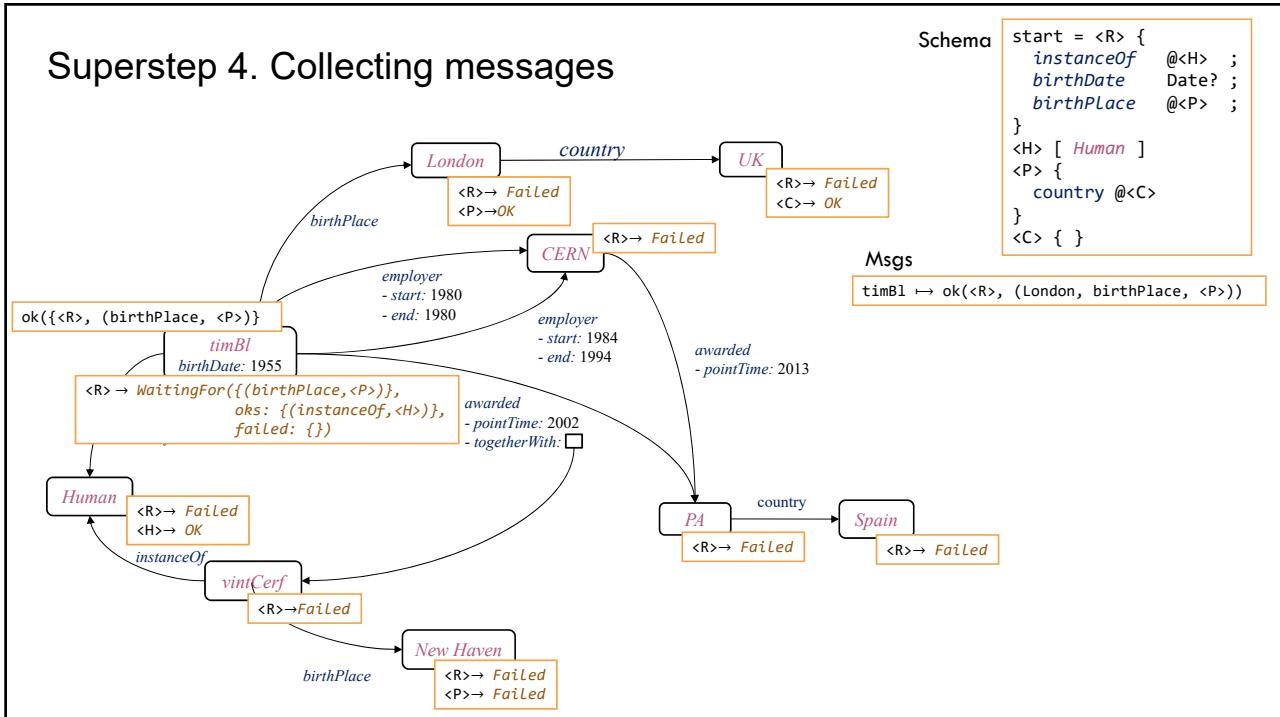
Msgs

London  $\mapsto$  ok(<P>, (UK, country, <C>))

### Superstep 3. After vprog



### Superstep 4. Collecting messages



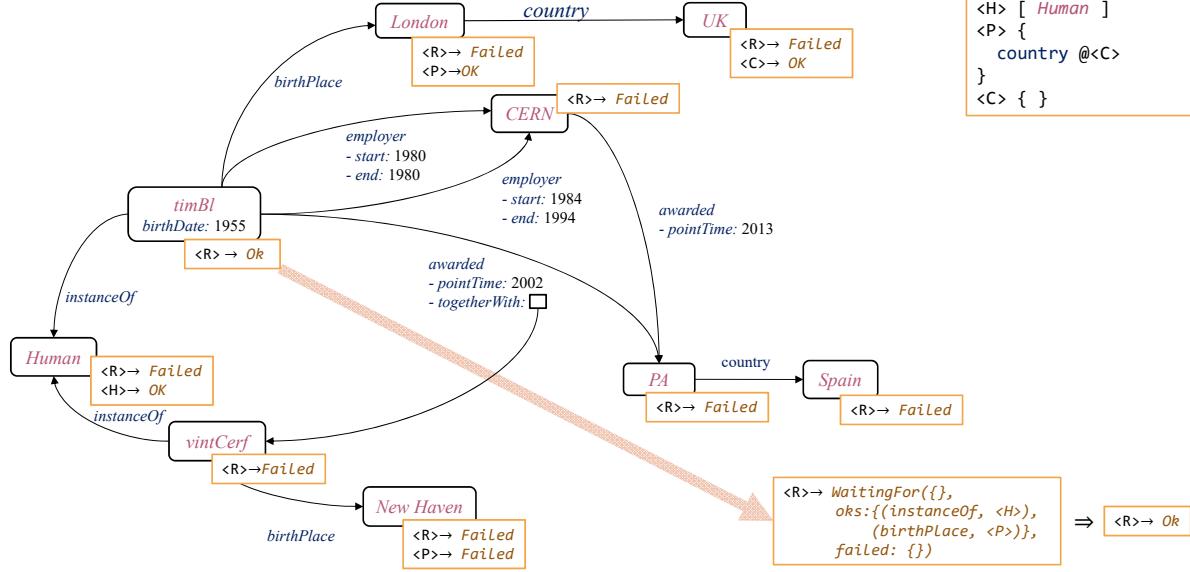
### Superstep 4. After vprog

Schema

```

start = <R> {
  instanceOf @<H> ;
  birthDate Date? ;
  birthPlace @<P> ;
}

<H> [ Human ]
<P> {
  country @<C>
}
<C> { }
  
```



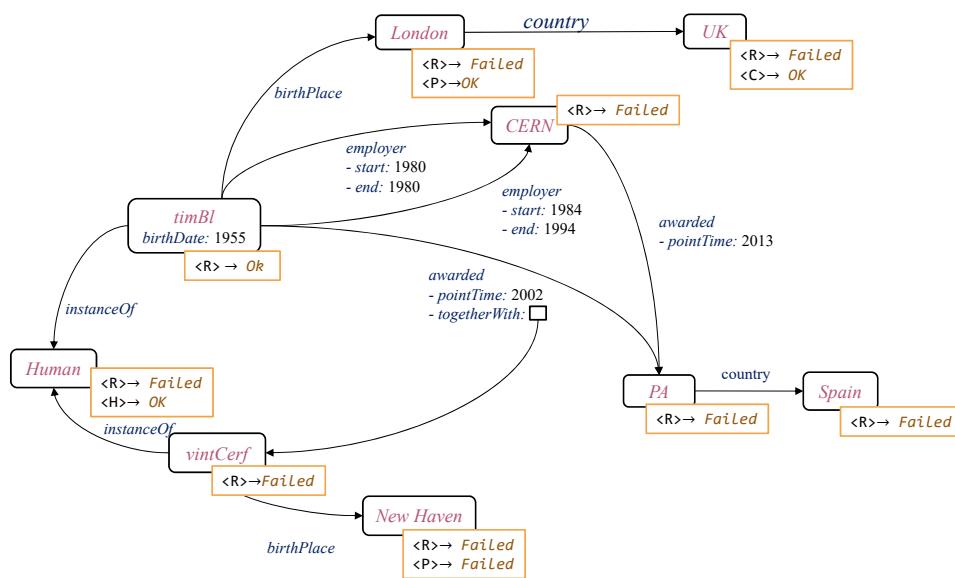
### End state: All nodes inactive, no more messages

Schema

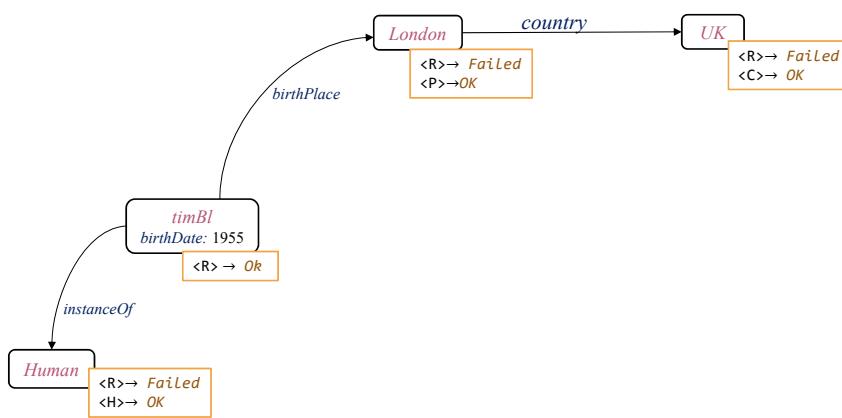
```

start = <R> {
  instanceOf @<H> ;
  birthDate Date? ;
  birthPlace @<P> ;
}

<H> [ Human ]
<P> {
  country @<C>
}
<C> { }
  
```



## Wikibase Subgraph: entities with some OK shape



## Scheme

```
start = <R> {
    instanceof @<H> ;
    birthDate Date? ;
    birthPlace @<P> ;
}
<H> [ Human ]
<P> {
    country @<C>
}
<C> { }
```

# Status and Messages

<i>Status</i>	$::=$	<i>Undefined</i>	Default status
		<i>Ok</i>	Node conforms
		<i>Failed</i>	Node doesn't conform
		<i>Pending</i>	Requested to conform
		<i>WaitingFor(ds, oks, fs)</i>	Waiting for some neighbours
		<i>ds</i> = list dependants neighbours	
		<i>oks</i> = list of conformant neighbours	
		<i>fs</i> = list of non conformant neighbours	
			where <i>ds, oks, failed</i> $\in \mathcal{V} \times \mathcal{P} \times \mathcal{L}$

<b>Msg</b>	<b> ::= </b>	<i>Validate</i>	Request to validate
		<i>Checked(oks, fs)</i>	Some neighbours have been checked <i>oks</i> = neighbours that have been checked as conformant <i>fs</i> = neighbours that have been checked as non-conformant
		<i>WaitFor(ds)</i>	Request to wait for some neighbours where $ds \in \mathcal{V} \times \mathcal{P} \times \mathcal{L}$

# Adapting Pregel to validate graphs using ShEx

**Algorithm 1:** Pregel-based ShEx validation pseudocode

**Input parameters:**

```

g: Graph[V, E]
initialLabel: L
checkLocal: (L, V) → Ok| Failed| Pending(Set[L])
checkNeighs: (L, Bag[(E, L)], Set[(E, L)]) → Ok| Failed
tripleConstraints: L → Set[(E, L)]
Output: g:Graph[(V, L ↦ Status), E]
gs = mapVertices(g, λ(id, v) → (id, (v, λv → Undefined)))
gs = pregel(Validate, gs, vProg, sendMsg, mergeMsg)
gs = mapVertices(gs, checkUnsolved)
return gs
def checkUnsolved(v,m) = (v,m') where
  m'(l) =
  { checkNeighs(l, ∅, ∅) if m(l) = Pending
    checkNeighs(l, oks, fs ∪ ds) if m(l) = WaitingFor(ds, oks, fs)
    m(l) otherwise
  }

```

**Parameters**

InitialLabel, start label  
checkLocal, attempts to check if a node conforms locally. Possible results:

Ok

Failed

Pending(ls)

checkNeighs, attempts to check the neighbourhood of a node. Result:

Ok

Failed

tripleConstraints: set of constraints associated with a label

## vProg definition

$vProg(id, (n, m), msg) = (n, m')$  where  
 $m'(l) = m(l)$

except for the cases indicated by the rules:

$$\frac{(n, m), l \rightsquigarrow Validate \quad m(l) = s \in \{Undefined, Pending\} \quad checkLocal(l, n) = r \in \{Ok, Failed\}}{m'(l) = r}$$

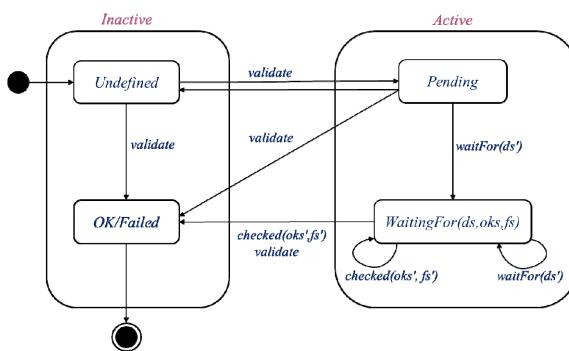
$$\frac{(n, m), l \rightsquigarrow Validate \quad m(l) = r \in \{Undefined, Pending\} \quad checkLocal(l, n) = Pending(ls)}{\begin{aligned} m'(l) &= Undefined \\ m'(l') &= Pending \quad \forall l' \in ls \end{aligned}}$$

$$\frac{(n, m), l \rightsquigarrow Validate \quad m(l) = r \in \{Ok, Failed\}}{m'(l) = r}$$

$$\frac{(n, m), l \rightsquigarrow Validate \quad m(l) = WaitingFor(ds, oks, fs)}{m'(l) = Ok}$$

$$\frac{(n, m), l \rightsquigarrow Checked(oks, fs) \quad ds \setminus (oks \cup fs) \neq \emptyset}{\begin{aligned} m(l) &= WaitingFor(ds, oks', fs') \\ m'(l) &= WaitingFor(ds, oks \cup oks', fs \cup fs') \end{aligned}}$$

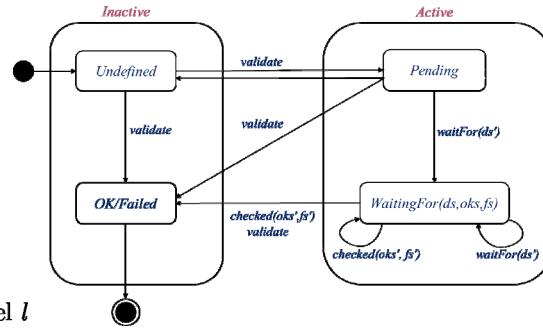
$$\frac{(n, m), l \rightsquigarrow Checked(oks, fs) \quad ds \setminus (oks \cup fs) = \emptyset}{\begin{aligned} m(l) &= WaitingFor(ds, oks', fs') \\ m'(l) &= checkNeighs(l, oks \cup oks', fs \cup fs') \end{aligned}}$$



## sendMsg

$\langle (s, m_s), p, (o, m_o) \rangle =$   
 triplet view with  
 - subject  $(s, m_s)$   
 - predicate  $p$   
 - object  $(o, m_o)$

$(x, m_x), l \rightsquigarrow Msg =$   
 message  $Msg$  sent to node  $x$  with status map  $m_x$  for label  $l$



$$\frac{\langle (s, m_s), p, (o, m_o) \rangle \in \mathcal{G} \quad m_s(l) = Pending \quad tcs(l, \mathcal{S}) = \cup \xrightarrow{p} @l'}{(s, m_s), l \rightsquigarrow WaitFor((o, p, l'))}$$

$(o, m_o), l \rightsquigarrow Validate$

$$\frac{\langle (s, m_s), p, (o, m_o) \rangle \in \mathcal{G} \quad m_s(l) = WaitingFor(ds, oks, fs) \quad (o, p, l') \in ds \quad m_o(l') = Ok}{(s, m_s), l \rightsquigarrow Checked((o, p, l'), \emptyset)}$$

$$\frac{\langle (s, m_s), p, (o, m_o) \rangle \in \mathcal{G} \quad m_s(l) = WaitingFor(ds, oks, fs) \quad (o, p, l') \in ds \quad m_o(l') = Failed}{(s, m_s), l \rightsquigarrow Checked(\emptyset, (o, p, l'))}$$

## mergeMsg

$$\text{mergeMsg}((n, m), l \rightsquigarrow msg_1, (n, m), l \rightsquigarrow msg_2) = (n, m), l \rightsquigarrow msg_1 \oplus msg_2$$

$Validate \oplus y$ $Validate \oplus Checked(oks, fs)$ $Validate \oplus WaitFor(ds)$ $Checked(oks, fs) \oplus Validate$ $Checked(oks, fs) \oplus Checked(oks', fs')$ $Checked(oks, fs) \oplus WaitFor(ds)$ $WaitFor(ds) \oplus Validate$ $WaitFor(ds) \oplus Checked(oks, fs)$ $WaitFor(ds) \oplus WaitFor(ds')$	$= y$ $= Checked(oks, fs)$ $= WaitFor(ds)$ $= Checked(oks, fs)$ $= Checked(oks \cup oks', fs \cup fs')$ $= Checked(oks \cup ds, fs \cup fs)$ $= WaitFor(ds)$ $= Checked(oks \cup ds, fs)$ $= WaitFor(ds \cup ds')$
--	--

## checkLocal

$\text{checkLocal}: (\mathcal{L}, \mathcal{V}) \rightarrow \text{Ok} | \text{Failed} | \text{Pending}(\text{Set}[\mathcal{L}])$

Checks if it is possible to validate a node locally

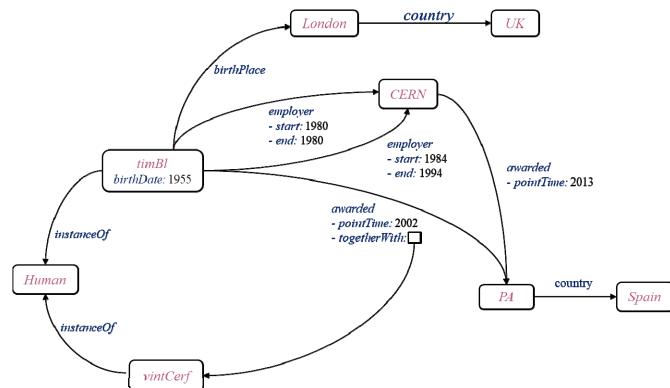
Returns

*Ok*: Node conforms

*Failed*: Node doesn't conform

*Pending*(*ls*) conformance depends on neighbours

```
checkLocal(<Human>, Human) = Ok
checkLocal(<Researcher>, Human) = Failed
checkLocal(<Researcher>, timBL) = Pending { <Human>, <Place> }
```



```
start = <Researcher>
<Researcher> {
  instanceOf @<Human> ;
  birthDate Date? ;
  birthPlace @<Place> ;
}
<Human> [ Human ]
<Place> {
  country @<Country>
}
<Country> { }
```

## checkNeighs

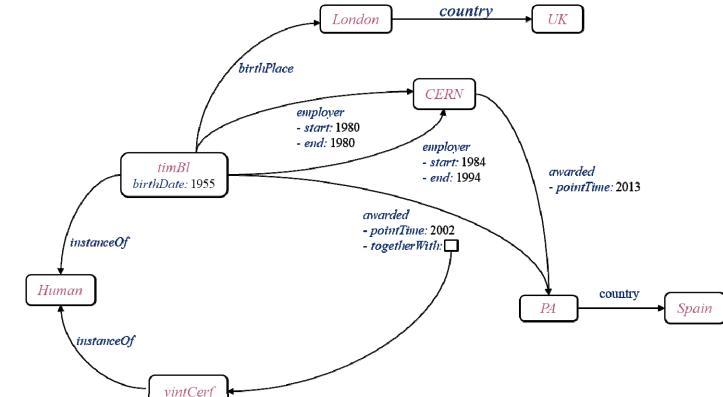
$\text{checkNeighs}: (\mathcal{L}, \text{Bag}[(\mathcal{E}, \mathcal{L})], \text{Set}[(\mathcal{E}, \mathcal{L})]) \rightarrow \text{Ok} | \text{Failed}$

Checks if the bag of neighbours of a node conform to the regular bag expression defined by a shape

Returns: ok (conforms), failed (doesn't)

```
checkNeighs(<Researcher>, {<instanceOf>, <Human>}, {<birthPlace>, <London>}) = Ok
checkNeighs(<Researcher>, {<instanceOf>, <Human>}, {}) = Failed
```

```
rbe(<Researcher>) = (<instanceOf>, <Human>);
(birthPlace, <Place>)
```



```
start = <Researcher>
<Researcher> {
  instanceOf @<Human> ;
  birthDate Date? ;
  birthPlace @<Place> ;
}
<Human> [ Human ]
<Place> {
  country @<Country>
}
<Country> { }
```

## tripleConstraints

tripleConstraints:  $\mathcal{L} \rightarrow \text{Set}[(\mathcal{E}, \mathcal{L})]$

Returns the triple constraints of a shape label

```
tripleConstraints(<Researcher>) = {(instanceOf, <Human>), (birthPlace, <Place>)}
tripleConstraints(<Human>) = {}
tripleConstraints(<Place>) = {(country, <Country>)}
tripleConstraints(<Country>) = {}
```

```
start = <Researcher>
<Researcher> {
  instanceOf @<Human> ;
  birthDate Date? ;
  birthPlace @<Place> ;
}
<Human> [ Human ]
<Place> {
  country @<Country>
}
<Country> { }
```

## Preliminary results

Implemented in SparkWDSUB using Apache Spark GraphX

No optimizations applied

Latest Wikidata dumps on AWS can be processed on AWS

512 cores, 3.904 Gb RAM, 121.600 Gb disk

For 2014 Wikidata dump (31.3 GB uncompressed) in 3 minutes

For 2021 Wikidata dump (1.256,55 Gb uncompressed) it took 36 minutes

Only for simple schemas yet

# Conclusions

Formal definition of Knowledge graphs  
 RDF graphs, property graphs, wikibase graphs

ShEx extension for property graphs/wikibase graphs

Formal definition of wikibase graphs subsets

Scalable approach to validate big knowledge graphs

Inspired by Pregel algorithm (Spark GraphX)

# Repaso de contribuciones

	Grafos RDF	Property graphs	Grafos Wikibase
Definición formal	Realizado previamente	Realizado previamente	Presente trabajo
Descripción grafos conocimiento	ShEx Realizado anteriormente	PShEx Presente trabajo	WShEx Presente trabajo
Entity-generated (definición)	-	-	Presente trabajo
Entity-generated (implementación)	-	-	WDumper
Simple matching (definición)	-	-	Presente trabajo
Simple matching (implementación)	-	-	Wdumper
ShEx-based matching (definición)	-	-	Presente trabajo
ShEx-based matching (implementación)	-	-	WDSub
ShEx+Slurp (definición)	-	-	Presente trabajo
ShEx+Slurp (implementación)	ShEx.js, pyShEx	-	-
ShEx+Pregel (definición)	-	-	Presente trabajo
ShEx+Pregel (implementación)	-	-	SparkWDSub

# Fin presentación prueba 2

Jose Emilio Labra Gayo



Universidad de Oviedo