EA-to-RDF: Pain-Free integration of RDF and UML*

Bob Janssen[1,*,†], Ghislain Atemezing[2,†]

[1]*Joint Informatievoorziening Commando, Ministry of Defense, Netherlands*
[2]*ERA - European Union Agency for Railways, 120 Rue Marc Lefrancq, 59307 Valenciennes, France*

Abstract

This article presents an addon to the popular Enterprise Architect (EA) tool that lets modellers import existing ontologies into EA as classes and export class models to ontologies. The addon implements the SEMIC guidelines that standardise the UML-RDF mapping. The tool has been tested against some large UML diagrams from the railway domain and the generated outputs (RDF and UML) demonstrate the usefulness of EA-RDF both from UML experts and ontology developers. This approach bridges the gap between ontologies and IT engineering in UML; modellers at ease in UML but unfamiliar with triples can now easily access existing ontologies and publish their models as triples.

Keywords

UML diagrams, RDF, OWL, Ontology, Enterprise Architect, Model Based Systems Engineering

1. Introduction

Engineers and ontologists share a desire to capture knowledge of systems. Engineers prefer graphic schemata whilst ontologists prefer language tooling. The common goal remains the need to precisely model systems. Whether it is a technical control-and-command system or a socio-economic system, modellers apply the divide-and-conquer approach to identify the components and subcomponents that form the system and how they relate.

Information technologists, better known under the acronym IT-engineers, use the Unified Modelling Language (UML) [1], and in particular class models, to capture system information. The UML is a rich graphical language that lets engineers sketch diagrams showing concepts and how they connect. Ontologists are more philosophically and linguistically inclined but in essence do the same when they identify concepts and how they relate in subject-predicate-object phrases.

Good IT design separates problem and solution. Early design in particular should avert technical products that may solve a problem. This is important because IT design all to often is burdened with legacy technologies that hinder sound conceptualisation. Instead, IT engineering should exert intellectual rigour by eliciting use cases, and then codifying the discipline's jargon in terms of UML class diagrams *before* tying themselves to a given technology. Introducing

Semantics2024: 20th International Conference on Semantic Systems, September 17–19, 2024, Amsterdam, The Netherlands

[*] Corresponding author.

[†] These authors contributed equally.

bob.janssen@yahoo.com (B. Janssen); ghislain.atemezing@era.europa.eu (G. Atemezing)
https://ijzerweg.nl (B. Janssen); https://atemezing.org (G. Atemezing)
0000-xxxx-xxxx-xxxx (B. Janssen); 0000-0003-1562-6922 (G. Atemezing)

technological platforms or ”code smell” into a model is considered poor practice because technologies come and go whereas concepts last.

UML class diagrams are designed to capture concepts and by this virtue are a favorite tool with both IT engineers and ontologists. This raises the question how information can cross more easily from UML to triples and back. IT engineers have much to win from reusing the knowledge that is readily available in published ontology. And vice versa, knowledge engineers can raise interest in ontologies with IT engineers.

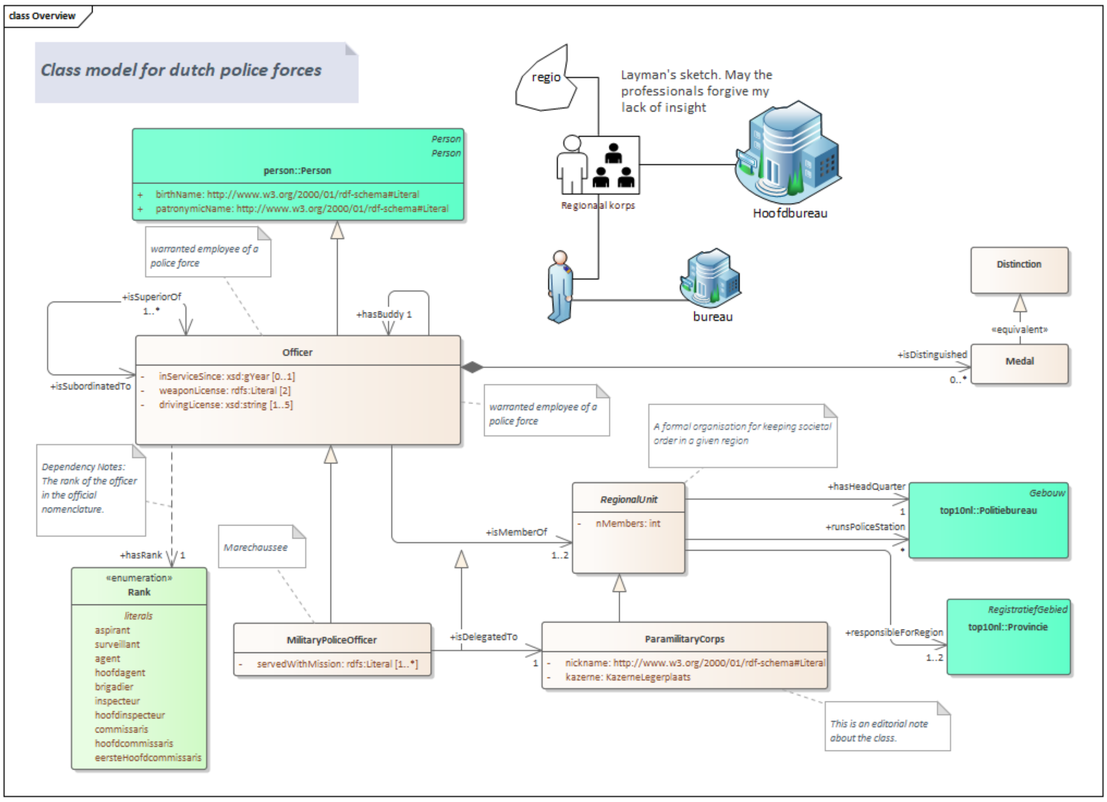

class Overview
Class model for dutch police forces
regio
Layman's sketch. May the professionals forgive my lack of insight
Regionaal korps
Hoofdbureau
bureau
Person
Person
person::Person
+ birthName: http://www.w3.org/2000/01/rdf-schema#Literal
+ patronymicName: http://www.w3.org/2000/01/rdf-schema#Literal
warranted employee of a police force
+isSuperiorOf
1..*
+hasBuddy 1
+isSubordinatedTo
Officer
- inServiceSince: xsd:gYear [0..1]
- weaponLicense: rdfs:Literal [2]
- drivingLicense: xsd:string [1..5]
warranted employee of a police force
Distinction
«equivalent»
+isDistinguished
0..*
Medal
A formal organisation for keeping societal order in a given region
Dependency Notes: The rank of the officer in the official nomenclature.
+hasRank
1
«enumeration»
Rank
literals
aspirant
surveillant
agent
hoofdagent
brigadier
inspecteur
hoofdinspecteur
commissaris
hoofdcommissaris
eersteHoofdcommissaris
Marechaussee
+isMemberOf
1..2
RegionalUnit
- nMembers: int
+hasHeadQuarter
1
+runsPoliceStation
*
Gebouw
top10nl::Politiebureau
MilitaryPoliceOfficer
- servedWithMission: rdfs:Literal [1..*]
+isDelegatedTo
1
ParamilitaryCorps
- nickname: http://www.w3.org/2000/01/rdf-schema#Literal
- kazerne: KazerneLegerplaats
+responsibleForRegion
1..2
RegistratiefGebied
top10nl::Provincie
This is an editorial note about the class.

Figure 1: This class diagram exemplifies reuse of imported ontologies, the W3C person ontology and the Dutch land registry’s top10nl ontology that defines topographic features. The diagram closes the gap between IT- and domain expert by mixing a figure with explanations.

The SEMIC work group published a standard for mapping a small subset of UML constructs to triples and back. We show in this paper how this mapping contributed to tooling and a workflow for the widely used UML tool Enterprise Architect (EA). Section 2 of this paper presents existing approaches, section 3 presents the main functionalities of the tool.

2. Related Work

Some approaches have been proposed in recent years to bridge the gap between UML modeling and OWL ontologies. The work presented in [2] is an early attempt to generat OWL from UML models. The approach used XMI corresponding to Ontology UML Profile (OUP) into OWL description. Unfortunately, it was not possible to access and assess the tool.

Another approach by Chavez-Feria and al., [3] proposed Chowlk, a framework to transform digital ontology conceptualisation diagrams into OWL. Chowlk proposed a different UML-based visual notation and templates based on diagrams.net, which is not necessary in our proposal. Finally, the OSLO project [4] boosted the next generation of formal mappings of UML to OWL.

The OMG ontology workgroup, having identified the need for converting UML to RDF, created the Ontology Definition Metamodel (ODM) [5]. The ODM is included in EA so modellers wishing to publish an ontology apply ODM stereotypes to classes and relations. But the need to apply stereotypes expresses the intent to publish RDF from the outset which may not be given when modellers wish to steer clear of any "platform" whether it be XSD, Json, Object Oriented code, or RDF.

The SEMIC guidelines [6] (mostly) ignore stereotypes and apply to a subset of UML class models. This Platform Independent Model (PIM) approach expresses the intent that a good model should be like a stem cell that is pluripotent and later develops into Platform Specific Model (PSM) whence into even more specific computer code.

The SEMIC workgroup also produced a tool based on XSLT that turns a class model, expressed in XMI, into RDF. Whilst this is tool-independent, XSLT is not particularly user friendly. The present addOn for EA was developed to be user friendly.

3. EA-RDF Functionalities

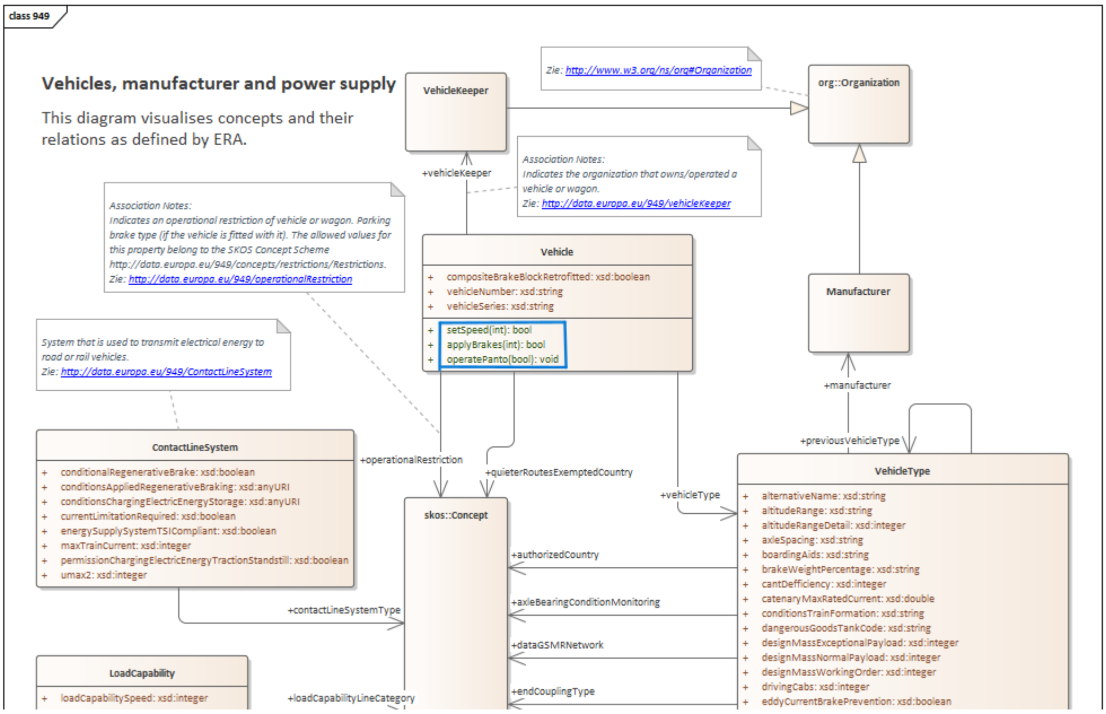
In this section, we describe how to go from RDF to UML and vice versa.

3.1. RDF to UML

Published ontologies are potent stores of domain knowledge. In day-to-day life, most IT engineers reinvent the wheel or at best take inspiration from ontologies when designing class diagrams because they're tasked to build an application on a given platform. This defies the Interoperability and Reusability goals in the FAIR acronym. This motivated the capability of importing published ontologies into EA. An IT engineer can climb on the shoulders of giants by importing ontologies into UML classes. The import retains important information including definitions, the URI's and relations. The tool comes with a set of preloaded well-known ontologies such as `foaf`, DCAT, SKOS and many more. Alternatively, the user can import published ontologies straight from existing catalogs such as Linked Open Vocabulary (LOV) or any resource from the web. The modeller can then tailor the imports by specialising classes, adding relations and new classes. This prevents modellers from toiling away in isolation and by design create incompatibilities for which costly and fragile adaptors need to be built as an afterthought. Short, importing ontologies as ready-made UML class models gives Model Base System Engineers a sound foundation and blueprints on which to build models for their domain.

3.2. UML to RDF

The SEMIC style guide reference [6] lists rules for mapping classes, data types, connectors, arities and annotations to RDF concepts, properties and SHACL shapes. UML constructs not in



Vehicles, manufacturer and power supply

This diagram visualises concepts and their relations as defined by ERA.

Zie: http://www.w3.org/ns/org#Organization

Association Notes:
Indicates the organization that owns/operated a vehicle or wagon.
Zie: http://data.europa.eu/949/vehicleKeeper

Association Notes:
Indicates an operational restriction of vehicle or wagon. Parking brake type (if the vehicle is fitted with it). The allowed values for this property belong to the SKOS Concept Scheme http://data.europa.eu/949/concepts/restrictions/Restrictions.
Zie: http://data.europa.eu/949/operationalRestriction

System that is used to transmit electrical energy to road or rail vehicles.
Zie: http://data.europa.eu/949/ContactLineSystem

class 949
VehicleKeeper
org::Organization
+vehicleKeeper
Vehicle
+ compositeBrakeBlockRetrofitted: xsd:boolean
+ vehicleNumber: xsd:string
+ vehicleSeries: xsd:string
+ setSpeed(int): bool
+ applyBrakes(int): bool
+ operatePanto(bool): void
Manufacturer
+manufacturer
+previousVehicleType
ContactLineSystem
+ conditionalRegenerativeBrake: xsd:boolean
+ conditionsAppliedRegenerativeBraking: xsd:anyURI
+ conditionsChargingElectricEnergyStorage: xsd:anyURI
+ currentLimitationRequired: xsd:boolean
+ energySupplySystemTSICompliant: xsd:boolean
+ maxTrainCurrent: xsd:integer
+ permissionChargingElectricEnergyTractionStandstill: xsd:boolean
+ umax2: xsd:integer
+operationalRestriction
+quieterRoutesExemptedCountry
+vehicleType
skos::Concept
VehicleType
+ alternativeName: xsd:string
+ altitudeRange: xsd:string
+ altitudeRangeDetail: xsd:integer
+ axleSpacing: xsd:string
+ boardingAids: xsd:string
+ brakeWeightPercentage: xsd:string
+ cantDefficiency: xsd:integer
+ catenaryMaxRatedCurrent: xsd:double
+ conditionsTrainFormation: xsd:string
+ dangerousGoodsTankCode: xsd:string
+ designMassExceptionalPayload: xsd:integer
+ designMassNormalPayload: xsd:integer
+ designMassWorkingOrder: xsd:integer
+ drivingCabs: xsd:integer
+ eddyCurrentBrakePrevention: xsd:boolean
+authorizedCountry
+axleBearingConditionMonitoring
+dataGSMRNetwork
+endCouplingType
+contactLineSystemType
LoadCapability
+ loadCapabilitySpeed: xsd:integer
+loadCapabilityLineCategory

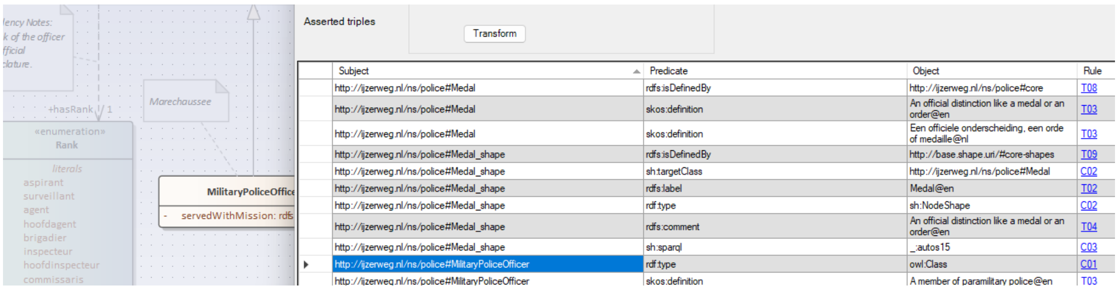
Figure 2: Screenshot of a diagram showing a selection of classes imported from an ontology defined by the European Railway Agency (ERA). A user typically creates such a view to represent a domain of interest and then enriches the original model, e.g. by adding methods as shown by the blue square in the Vehicle class, or by specialising classes to meet specific requirements.

the mapping rules are ignored.

URI's are essential to triples so the UML model must provide the ingredients for constructing URI's. This is commonly achieved by applying tags named "uri" to UML elements or "base-uri" to packages. This closely mimicks the SEMIC approach and dispenses of the need for a separate configuration file. A subject, predicate or object node will inherit the URI from the tag of the original UML element or, lacking a tag, of the package where it was found. The tool issues a warning or error when no URI could be constructed.

3.3. Publishing your model to the Web

A model that is published on the web is FAIR: Findable - Accessible - Interoperable and Reusable. EA out-of-the-box can produce a HTML view of a UML model. The addon adds a new landing page that lists the produced RDF-triples. This HTML is readily exposed on a web server in such a way that the produced URI's have a target on the the web. Users can browse context information because the triples are backed by published UML diagrams. Extra context reduces entry barriers, accelerates learning curves and reduces development effort. The landing page includes an index that helps users discover concepts defined in the model. This creates a continuum where one can navigate between triples and published UML. This compares favourably to a classic



Subject	Predicate	Object	Rule
http://ijzerweg.nl/ns/police#Medal	rdfs:isDefinedBy	http://ijzerweg.nl/ns/police#core	T08
http://ijzerweg.nl/ns/police#Medal	skos:definition	An official distinction like a medal or an order@en	T03
http://ijzerweg.nl/ns/police#Medal	skos:definition	Een officiele onderscheiding, een orde of medaille@nl	T03
http://ijzerweg.nl/ns/police#Medal_shape	rdfs:isDefinedBy	http://base.shape.uri/#core-shapes	T09
http://ijzerweg.nl/ns/police#Medal_shape	sh:targetClass	http://ijzerweg.nl/ns/police#Medal	C02
http://ijzerweg.nl/ns/police#Medal_shape	rdfs:label	Medal@en	T02
http://ijzerweg.nl/ns/police#Medal_shape	rdf:type	sh:NodeShape	C02
http://ijzerweg.nl/ns/police#Medal_shape	rdfs:comment	An official distinction like a medal or an order@en	T04
http://ijzerweg.nl/ns/police#Medal_shape	sh:sparql	_:autos15	C03
http://ijzerweg.nl/ns/police#MilitaryPoliceOfficer	rdf:type	owl:Class	C01
http://ijzerweg.nl/ns/police#MilitaryPoliceOfficer	skos:definition	A member of paramilitary police@en	T03

Figure 3: During the transformation process a table displays the produced triples. A click on a triple highlights the associated element, in this case the MilitaryPoliceOfficer class. The right-most column links to the SEMIC rule that governed the creation of this triple. This helps understanding and verifying the transformation process.

situation where users chase data, context and documentation locked away in various silos.

4. Conclusions and future work

This paper presents a tool for automatic generation of RDF from UML class models and vice-versa, packaged as an add-on for Enterprise Architect, a popular UML modelling tool. The tool implements the SEMIC mapping rules. We will perform a quantitative evaluation of the tool by both UML modellers and ontologists to compare the tool with existing approaches.

References

[1] O. A. Specification, Omg unified modeling language (omg uml), superstructure, v2. 1.2, Object Management Group 70 (2007) 16.

[2] D. Gasevic, D. Djuric, V. Devedzic, V. Damjanovi, Converting uml to owl ontologies, in: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, 2004, pp. 488–489.

[3] S. Chávez-Feria, R. García-Castro, M. Poveda-Villalón, Converting uml-based ontology conceptualizations to owl with chowlk, in: The Semantic Web: ESWC 2021 Satellite Events, Springer International Publishing, Cham, 2021, pp. 44–48.

[4] D. De Paepe, G. Thijs, R. Buyle, R. Verborgh, E. Mannens, Automated uml-based ontology generation in oslo2, in: E. Blomqvist, K. Hose, H. Paulheim, A. Ławrynowicz, F. Ciravegna, O. Hartig (Eds.), The Semantic Web: ESWC 2017 Satellite Events, Springer International Publishing, Cham, 2017, pp. 93–97.

[5] O. M. Group, Ontology definition metamodel, 2014. URL: https://www.omg.org/spec/ODM/1.1/PDF.

[6] E. Costetchi, B. V. Nuffelen, C. Nyulas, E. Stani, A.-N. Pasare, Semic style guide – technical guidelines, 2023. URL: https://semiceu.github.io/style-guide/1.0.0/references.html, v. 1.0.0.