DLs	Protégé	Owlready	OWLAPI

			OWLOntologyManager m = OWLManager.createOWLOntologyManager();
			OWLDataFactory df = OWLManager.getOWLDataFactory();
			OWLOntology o = m.createOntology(MY_IRI);

A ⊑ B	A subclass of B	class A(B): ... (or) A.is_a.append(B)	m.applyChange(new AddAxiom(o, df.getOWLSubClassOfAxiom(A, B)));
A ⊓ B	A and B	A & B	df.getOWLObjectIntersectionOf(A, B)
A ⊔ B	A or B	A | B	df.getOWLObjectUnionOf(A, B)
¬ A	not A	Not(A)	df.getOWLObjectComplementOf(A)
A ⊓ B = ∅	A disjoint with B	AllDisjoint([A, B])	m.applyChange(new AddAxiom(o, df.getOWLDisjointClassesAxiom(A, B)));
A ≡ B	A equivalent to B	A.equivalent_to.append(B)	m.applyChange(new AddAxiom(o, df.getOWLEquivalentClassesAxiom(A, B)));
{i, j,...}	{i, j,...}	OneOf([i, j,...])	df.getOWLObjectOneOf(i, j, ...)
∃R.B	R some B	R.some(B)	df.getOWLObjectSomeValuesFrom(R, B)
∀R.B	R only B	R.only(B)	df.getOWLObjectOnlyValuesFrom(R, B)
=2R.B	R exactly 2 B	R.exactly(2, B)	df.getOWLObjectExactCardinality(2, R, B)
∃R.{i}	R value i	R.value(i)	df.getOWLObjectHasValue(R, i)
∃ R.⊤ ⊑ A	R domain A	R.domain = [A]	m.applyChange(new AddAxiom(o, df.getOWLObjectPropertyDomainAxiom(R, A)));
⊤ ⊑ ∀R.B	R range B	R.range = [B]	m.applyChange(new AddAxiom(o, df.getOWLObjectPropertyRangeAxiom(R, B)));
S ≡ R^-	S inverse of R	S.inverse = R	m.applyChange(new AddAxiom(o, df.getOWLInverseObjectPropertiesAxiom(R, S)));
A(i)	i type A	i = A() (or) i.is_instance_of.append(A)	m.applyChange(new AddAxiom(o, df.getOWLClassAssertionAxiom(A, i)));
R(i, j)	i object property assertion j	i.R = j (R is functional) (or) i.R.append(j) (otherwise)	m.applyChange(new AddAxiom(o, df.getOWLObjectPropertyAssertionAxiom(R, i, j)));
R(i, n)	i data property assertion j	i.R = n (R is functional) (or) i.R.append(n) (otherwise)	m.applyChange(new AddAxiom(o, df.getOWLDataPropertyAssertionAxiom(R, i, n)));
A⊑∃R.{i} ∧ (∃R^-.A)(i)	-	A.R = i (R is functional) (or) A.R.append(i) (otherwise)	-

Table 3: Correspondence between DLs, Protégé notations, Owlready syntax and OWLAPI syntax. A and B are classes, R and S are properties, i and j are individuals, n is a literal. The last line of the table shows the Owlready syntax for asserting role-fillers as class attributes, and the corresponding assertion in DLs.