icas-ontology

This is the unified ICAS ontology designed to describe the abstraction of information-security related information as used by performers on the DARPA ICAS project.

Distribution

This ontology is subject to public release approval by DARPA and as such may not be distributed outside of the DARPA ICAS program except under the terms of a non-disclosure agreement.

A Brief Explanation of OWL and RDF

The unified ICAS ontology is a collection of OWL2 ontologies under the `http://www.invincea.com/ontologies/icas/1.0/` namespace. Each ontology covers a specific conceptual area; for instance all information related to users and user accounts is captured in the User ontology.

OWL2 is an ontology description language built on top of the Resource Description Framework (RDF). RDF describes a way of storing data that is different from the traditional table-based conception. RDF data consists of triples, and only triples; each triple, called a *statement* consists of a *subject*, *predicate*, and an *object*.

The *subject* represents a resource of some kind, the *predicate* a relation, and the *object* can either be a literal value or another resource.

OWL2 and the RDF Schema (RDFS) define a set of resources and properties that can be used to develop ontologies for RDF datasets.

Examples

This section contains some annotated excerpts from the ontologies as examples of how to read RDF and OWL. These examples are encoded in the Terse RDF Triple Langauge (Turtle), just like the ICAS ontology. This is intended only as an overview and omits many details

Turtle Primer

Recall that RDF consists only of *statements*, each of which is composed of a *subject*, a *predicate*, and an *object*. In the Turtle encoding, there is a shorthand creating multiple statements that uses the `;` to indicate continuation with the same
subject. The following example shows how to interpret this syntax.

```
<subject> <predicate> <object> ;
            <predicate> <object> .

<subject-2> <predicate-2> <object-2> .
```

UserAccount Class

This example depicts the UserAccount class from the User ontology. It contains five statements in the Turtle syntax, all of which share `:UserAccount` as the subject.

```
:UserAccount a owl:Class ;
    rdfs:comment "an individual set of credentials."@en ;
    rdfs:label "User"@en ;
    rdfs:subClassOf owl:Thing .
```

The first statement `:UserAccount a owl:Class` states that `:UserAccount` has the type `owl:Class`. `owl:Class` is used to define types of resources.

The next two statements provide labels and comments – essentially annotations – a about `:UserAccount`. The third statement is standard and simply indicates that a `:UserAccount` is something that exists in the world

Datatype Property

Datatype Properties describe the types of literal information that can be represented and connected to resources. The following excerpt describes the `:hasFullName` property, which represents the concept of a user's display name.

```
:hasFullName
    a owl:DatatypeProperty ;
    rdfs:comment "and extended name or description, used only for display purposes"@en ;
    rdfs:label "has full name"@en ;
    rdfs:domain :UserAccount ;
    rdfs:range xsd:string .
```

The first statement is similar to what we saw before, but this time it indicates that `:hasFullName` is a DatatypeProperty. The comment and label fulfill the same role as in the prior example.

The last two statements `:hasFullName rdfs:domain :UserAccount` and `:hasFullName rdfs:range xsd:string` represent constraints on this property. The former states that only `:UserAccounts` may be the *subject* of this property. The latter says that the literal object value of this property must be of type string.

Object Property

Object Properties describe relationships between resources. The example below captures the notion of group

membership for users in computing systems. Note the use of `rdfs:domain` and `rdfs:range` to constrain the valid *subject* and *objects* for this *predicate*.

```
:memberOfGroup
    a owl:ObjectProperty ;
    rdfs:domain :UserAccount ;
    rdfs:label "is member of Group"@en ;
    rdfs:range :Group ;
```

File Structure

Each of these sub-ontologies is written to a separate file that shares the same name as the ontology using the Terse RDF Triple Langauge (Turtle) encoding. Thus, the User ontology can be found in user.ttl and the Authentication ontology can be found in authentication.ttl.

Although each file focuses on a specific set of concepts, they are not isolated. At the top of each file, a series of namespaces describe other ontologies from which that sub-ontology uses concepts.

Tools for Reading OWL2 Ontology Files

While OWL2 ontology files are human readable, the Semantic Web community has also developed a wide variety of GUI based tools that can be used to read and edit ontology files. Links to a selection of these tools can be found on the Semantic Web Wiki