
Ontologia aplicațiilor mobile

Studenti: Melemciuc Marius-Constantin, Hasna Robert
Profesor: Monica Tătarâm

Context. Scop

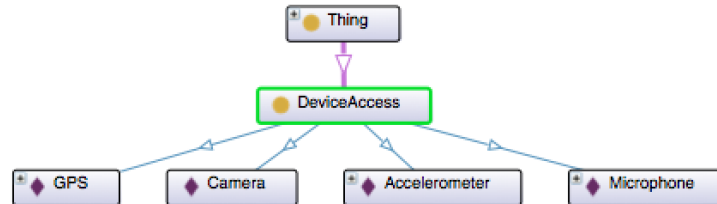
- Aplicațiile mobile reprezintă un exemplu sugestiv pentru elaborarea unei ontologii complexe și surprinderea legăturilor dintre modelările acestora.
 - Scopul ontologiei este acela de a oferi o privire cat mai generală, de ansamblu asupra modelării și clasificării aplicațiilor mobile, împreună cu niște exemple concrete.
-

Clase. Proprietăți

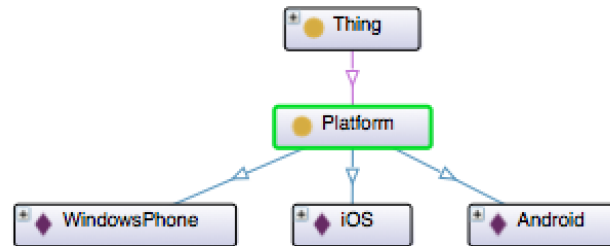
- Pentru crearea ontologiei s-au folosit 31 de clase, care desemnează cele mai de importante arii de interes pentru utilizatorii aplicațiilor mobile, care le reprezintă interesele, ceea ce urmăresc și după ce se ghidează ei.
 - Clasa *Categorie* reprezintă cea mai vastă și bine reprezentată clasă a ontologiei. Dintre subclase menționăm: *Business*, *Entertainment*, *Finance*, *Fitness*, *Games*, *Lifestyle*, *Social Networking*, *Utilities*, etc.
 - Fiecare exemplu concret de aplicație este reprezentat ca instanță a unei subclase din *Categorie*.
-

-
- Clasa *DeviceAccess* are rolul de a reprezenta facilitățile telefonului la care poate cere acces aplicația, sub forma cererii de drepturi.

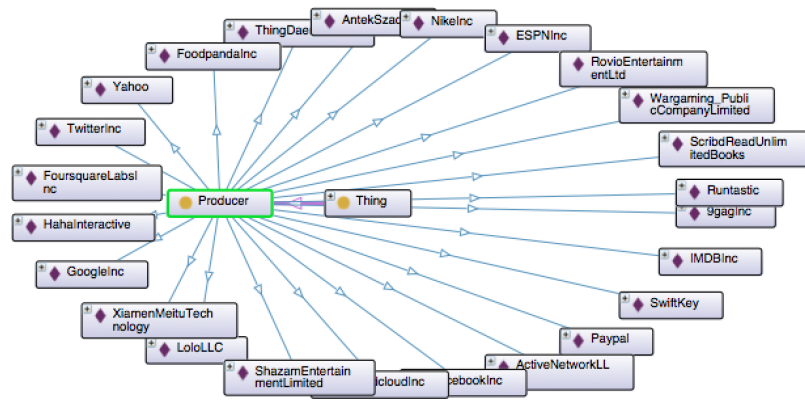
Ca subclase precizăm: *Accelerometru*, *Camera*, *GPS*, *Microfon*.



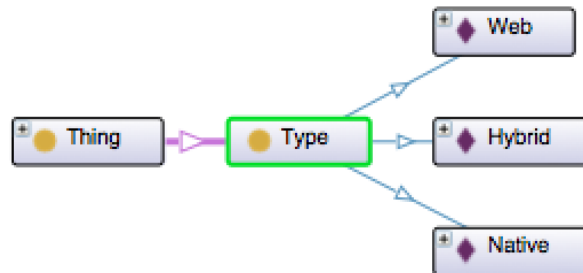
-
- Clasa *Platformă* reprezintă platformele mobile pentru care se pot dezolva aplicații, cum ar fi platforma *iOS* a celor de la Apple, *Android* a celor de la Google, și *WindowsPhone* a celor de la Microsoft.



- Clasa *Producător* modelează producătorii aplicațiilor mobile, entitățile de sine stătătoare care sunt responsabile de crearea, lansarea pe piață și întreținerea ulterioară în vederea bug-urilor.



-
- Clasa *Tip* se referă la cele trei tipuri importante din cadrul clasificării aplicațiilor mobile: *Native*, *Web*, *Hybrid*. Aplicațiile native reprezintă acele aplicații care sunt dezvoltate doar pentru telefoanele mobile, cele Web sunt cele care pot fi accesate sub forma de adresă URL, iar cele hibrid sunt aplicațiile cu suport de ambele părți.

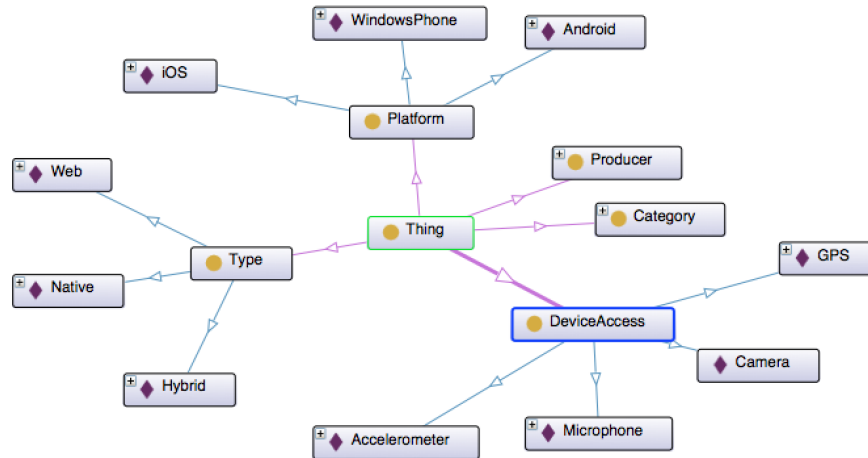


-
- Proprietățile in Protege:
 - proprietăți obiect
 - proprietăți de date
 - Primul tip de proprietăți sunt construite in jurul claselor. Se referă la diferite acțiuni raportate între clase:
 - *developedFor* - aplicația X dezvoltată pentru platforma Y
 - *isProducedBy* - aplicația X produsă de către Y
 - *isTypeOf* - aplicația X este de tipul Y
 - *requires* - aplicația X necesită permisiuni pentru a accesa funcțiile de Y ale telefonului
 - Proprietățile de date se referă la informațiile despre obiectele individuale.

Aceste proprietăți sunt esențiale pentru realizarea unui model viabil de ontologie si pentru interogări complexe:

 - *appID* - ID-ul aplicației
 - *name* - numele aplicației înregistrate
 - *producer_name* - numele producătorului/dezvoltatorului aplicației în cauză
 - *version* - versiunea platformei/sistemului de operare al telefonului mobil pe care ruleaza aplicația.
-

Diagrama entitate-relație



- Menționăm că clasele *Category* și *Producer* nu au mai fost expandate, fiind mult mai numeroase ca număr de subclase și ar fi încărcat prea mult diagrama.

Interogări

- O dată realizată ontologia, se pot rula interogări pe modelul creat pentru o mai bună înțelegere a clasificării.
 - Tab-ul DL Query oferă o modalitate simplă, intuitivă și ușor de utilizat pentru a căuta într-o ontologie clasificată pe mai multe niveluri.
-

DL Query pentru aflarea aplicațiilor dezvoltate pentru iOS, native, ce au nevoie de GPS.

The screenshot displays the 'Mobile_apps_ontology' web interface. The browser address bar shows the URL: http://www.semanticweb.org/ontologies/2015/2/Mobile_apps_ontology.owl. The interface includes tabs for 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Individuals', 'OWL Viz', 'DL Query', and 'OntoGraf'. The 'DL Query' tab is active.

Class hierarchy: DeviceAccess

- Thing
 - Category
 - DeviceAccess**
 - Platform
 - Producer
 - Type

Query:

Query (class expression)
Category and developedFor value iOS and isTypeOf value Native and requires value GPS

Execute Add to ontology

Query results

Sub classes (0)

Instances (9)

- NikeRunning
- Fancy
- GoogleMaps
- Foodpanda
- Foursquare
- CouchTo5K
- Runastic
- PerformanceStretching
- YahooWeather

☐ Super classes
☐ Ancestor classes
☐ Equivalent classes
☒ Subclasses
☐ Descendant classes
☒ Individuals

Reasoner active ☒ Show Inferences

Query pentru aflarea aplicațiilor native Android ce necesită acces la accelerometrul din cadrul telefonului.

The screenshot displays the Protégé OWL editor interface. The top navigation bar includes tabs for 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Individuals', 'OWL Viz', 'DL Query', and 'OntoGraf'. The 'DL Query' tab is selected.

On the left, the 'Class hierarchy: Type' panel shows a tree structure with 'Thing' as the root, followed by 'Category', 'DeviceAccess', 'Platform', and 'Producer'.

The main 'Query' panel contains a text area with the following query expression: `Category and developedFor value Android and isTypeOf value Native and requires value Accelerometer`. Below the text area are 'Execute' and 'Add to ontology' buttons.

The 'Query results' panel shows the following results:

- Sub classes (0)
- Instances (4)
 - NikeRunning
 - Foursquare
 - Runastic
 - WorldOfTanks

On the right side of the 'Query results' panel, there are checkboxes for filtering the results:

- ☐ Super classes
- ☐ Ancestor classes
- ☐ Equivalent classes
- ☒ Subclasses
- ☐ Descendant classes
- ☒ Individuals

At the bottom right, the status bar indicates 'Reasoner active' and 'Show Inferences' is checked.

Bibliografie

<http://protegewiki.stanford.edu/wiki/DLQueryTab>
