

University of Kentucky

Establishment of Student Recognition Awards and Prizes Memorandum

To: Ajay Sharma, Assistant Deputy Director of Financial Services at the University of Kentucky

From: Amol Shyamsunder Govekar, Hrishikesh Potdar, Jun Liu

Date: 5th December 2021

Subject: Reasoning with Protégé for Student Recognition Awards and Prizes.

We have created OWL ontology by Protégé. We created instance level assertions, defined data properties, and identified four inferences based on the ontology.

Explanation:

Classes:

1. AWARD: - AWARD are in the form of gifts or cash that may be made to the students at University of Kentucky by the university for the purpose for recognition. There are subclasses of CATEGORY1, CATEGORY2, CATEGORY3.
2. PROCUREMENT_CARD: - Used to purchase award.
3. PAYMENT REQUEST DOCUMENT: - Used to purchase award. Transaction id is attribute and a primary key.
4. UNIVERSITY_EMPLOYEE: - UNIVERSITY_EMPLOYEE includes all the employees of the university. It has two subclasses: APPROVING_OFFICIAL and NON_APPROVING_OFFICIAL and APPROVING_OFFICIAL has PRESIDENT and PROVOST
5. UNIVERSITY_FINANCIAL_SERVICES: - UFS approves procurement card.

Instances:

1. Award1: An instance of AWARD, a single award that is going to be awarded.
2. Award2: An instance of AWARD, a single award that is going to be awarded.
3. Award5: An instance of AWARD, a single award that is going to be awarded.
4. Award6: An instance of AWARD, a single award that is going to be awarded.
5. Award7: An instance of AWARD, a single award that is going to be awarded.
6. John: An instance of UNIVERSITY_EMPLOYEE, and an approving official.
7. Mary: An instance of UNIVERSITY_EMPLOYEE, and a cardholder.
8. Procurementcard1: An instance of PROCUREMENT_CARD.
9. Procurementcard2: An instance of PROCUREMENT_CARD.
10. Procurementcard3: An instance of PROCUREMENT_CARD.
11. Procurementcard4: An instance of PROCUREMENT_CARD.
12. Procurementcard5: An instance of PROCUREMENT_CARD.
13. UFS1: An instance of UNIVERSITY_FINANCIAL_SERVICES, approves procurement card.
14. ApprovingOfficial1: An instance of APPROVING_OFFICIAL.
15. Prd1: An instance of PAYMENT_REQUEST_DOCUMENT that purchases an award.

Object properties and Constraints:

1. approves: - It is an object property between the University Financial Services (domain constraint) and Procurement Card (range constraint).
2. approvedBy: - The Inverse of approves.
3. approvesReport: - It is an object property between the Approving Official (domain) and Award (range).
4. owns: - It is an object property between University Employee (domain) and Procurement Card (range).
5. ownedBy: - The Inverse of owns.
6. purchaseViaCard: - It is an object property between Procurement Card (domain) and Award (range).
7. purchaseByCard: Inverse of purchaseViaCard.
8. purchaseViaPrd: - It is an object property between Payment Request Document (domain) and Award (range)
9. submits: - It is an object property between the University Employee (domain) and Payment Request Document (range)

10. submittedBy: - The Inverse of submits.
11. purchaseByPrd: - Inverse of purchaseViaPrd.

Disjoint Constraint: All object properties have Disjointness constraint with each other.

Data properties

1. AWARD: - It has data properties like - awardId, purpose, category, sourceOfFunds, criteria, value, description, type, cardNoAward, prdTransactionId, purchaserType and approverId
2. PROCUREMENT_CARD: - cardNo, spendLimit, validityDuration, cardholderName, employeeIdPc, requestId
3. PAYMENT_REQUEST_DOCUMENT: - It has data properties like - transactionId, submitterId
4. UNIVERSITY_EMPLOYEE: - It has data properties like- employeeIdUe, employeeType, cardholdingStatus, departmentName
5. UNIVERSITY_FINANCIAL_SERVICES: - It has data properties like- requestIdUfs, requestType
6. APPROVING_OFFICIAL: - It has data properties like- approverType, employeeIdAo
7. NON_APPROVING_OFFICIAL: - It has data properties like- typeNpo, employeeIdNao

Explanation of Class Equivalences and their Inference

1. **AWARD:** purchasedByCard exactly 1 PROCUREMENT_CARD
Explanation: We set that an award can be purchased by only one PROCUREMENT_CARD
Individual: Award1; **Inference:** Type- CATEGORY1
Individual: Award2; **Inference:** Type- CATEGORY1
2. **CATEGORY2:** purchaseByPrd some PAYMENT_REQUEST_DOCUMENT
Explanation: We set that an award of CATEGORY2 can be purchased by any PAYMENT_REQUEST_DOCUMENT.
Individual: Award6; **Inference:** Type- CATEGORY2
3. **APPROVING_OFFICIAL:** approvesReport some AWARD
Explanation: We set that an APPROVING_OFFICIAL can approve an award.
Individual: John; **Inference:** Type- APPROVING_OFFICIAL
4. **PROVOST:** approvesReport some CATEGORY3
Explanation: We set that an PROVOST can approve CATEGORY3 award.
Individual: ApprovingOfficial1; **Inference:** Type- Provost
5. **UNIVERSITY_EMPLOYEE:** owns exactly 1 PROCUREMENT_CARD
Explanation: We set that a UNIVERSITY_EMPLOYEE can own only one PROCUREMENT_CARD
6. **PROCUREMENT_CARD:** ownedBy exactly 1 UNIVERSITY_EMPLOYEE
Explanation: We set that a PROCUREMENT_CARD can only be owned by one UNIVERSITY_EMPLOYEE.
7. **PAYMENT_REQUEST_DOCUMENT:** submittedBy exactly 1 UNIVERSITY_EMPLOYEE
Explanation: We set that a single PAYMENT_REQUEST_DOCUMENT can be submitted by only one UNIVERSITY_EMPLOYEE.
8. **CATEGORY1:** purchaseByCard some PROCUREMENT_CARD
Explanation: We set that an award of CATEGORY1 can be purchased by any PROCUREMENT_CARD.

Important Controversies in Ontology development and how we addressed it:

A controversy arose while designing data properties, we were not sure whether to add foreign keys present in table ("class" in context of ontology) as an independent data property. We resolved this confusion by discussing with TAs and removed redundant data properties added for every class.

Contributions of each group member: During our group meetings, all of us collectively created the OWL ontology with classes, instances, object properties, data properties, constraints, and class equivalences. Also, we collaboratively wrote the memo.

Amol S Govekar: Amol worked on creating the class equivalence and individuals.

Hrishikesh Potdar: Hrishikesh worked on deriving inferences rules based on object property constraints.

Jun Liu: Jun worked on drafting memo.