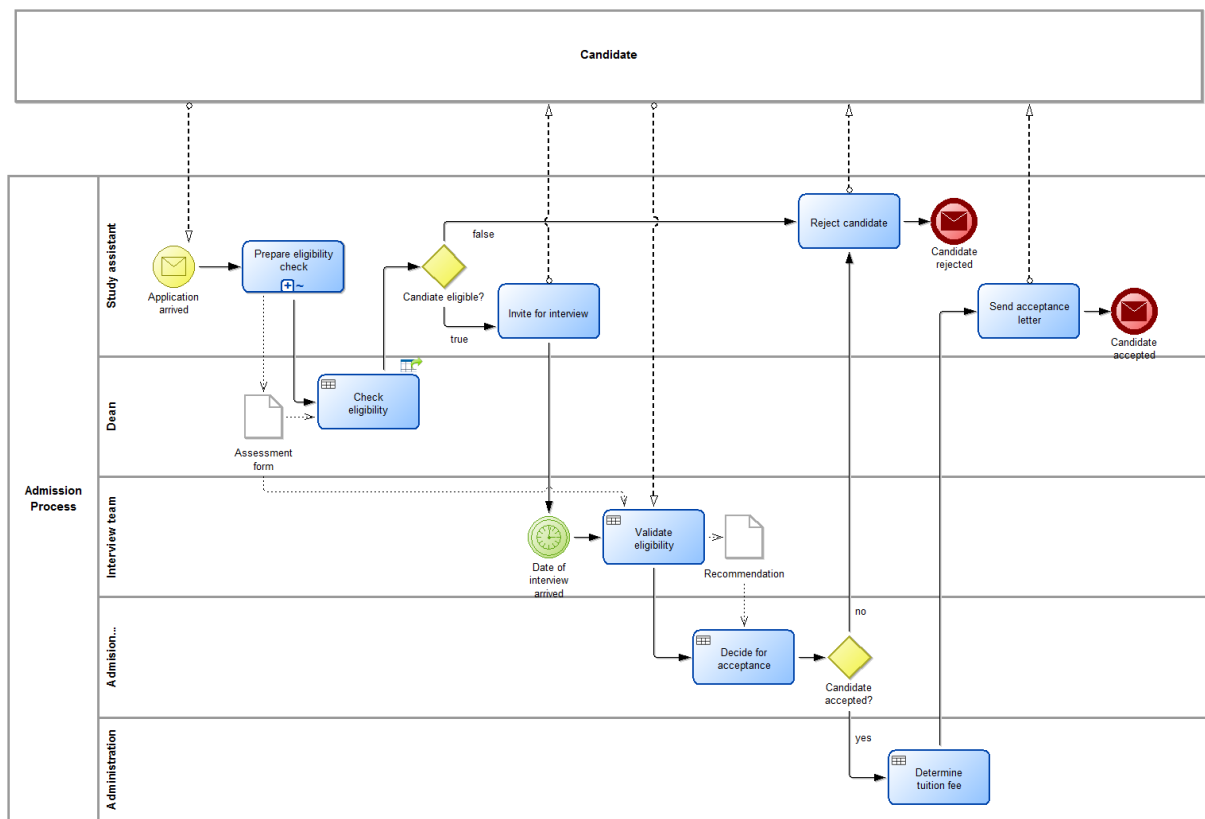


Exercise: Decisions in the Admission Process

This is still the model of the admission process to the Msc Business Information Systems.



Here are descriptions for the decisions to be taken in the three knowledge-intensive tasks:

Eligibility of the Candidate

Input for this task is the form filled by the study assistant:

MSc Business Information Systems

Candidate Profile

Name: _____

Bachelor Degree in: Information Systems
 Business Administration
 Information Technology
 other
 none

Grade: A B C D E

University: _____

Accreditation: yes no unclear

Months of professional experience: _____

Citizenship: Switzerland
 EU/EFTA
 other

The dean of study checks, whether the candidate is eligible. Candidates who do not have a bachelor degree from an accredited university are not eligible. The other candidates are invited for an interview, which is made by the interview team.

Acceptance of the Candidate

Input for this task is the form filled by the study assistant and additional clarifications from the interview, in particular whether the Bachelor degree is equivalent to Information Systems, Business Administration or Information Technology, and whether the grade and the professional experience are sufficient

Then the admission commission decides whether the candidate is accepted. *The candidate is accepted if she/he has a bachelor degree in Information Systems, Information Technology or Business Administration with at least good grade from an accredited university.*

Tuition Fee

For accepted candidates the administration determines the tuition fee: Swiss and European citizens pay CHF 700. The same is for Non-European students who are residents of Switzerland. Residency in Switzerland means that the person has residence permit C. Non-European students who are not residents of Switzerland pay CHF 7500.

Exercise

Decide these three decisions with rules (i.e. horn clauses)

Solution

Example Facts:

```
name(tim, "Sir Tim Berners-Lee").
bachelor(tim, other).
grade(tim, a).
university(tim, oxford).
accredited(oxford).
prof_experience(tim, 36).
citizenship(tim, eu).
```

Deciding Eligible

Variant 1:

```
bachelor_ok(X) :- bachelor(X, is).
bachelor_ok(X) :- bachelor(X, ba).
bachelor_ok(X) :- bachelor(X, it).
bachelor_ok(X) :- bachelor(X, other).

accreditation_ok(X) :- university(X, U), accredited(U).

eligible_ok(X) :- bachelor_ok(X), accreditation_ok(X).
```

Variant 2 (with negation)

```
bachelor_ok(X) :- not bachelor(X, none).

accreditation_ok(X) :- university(X, U), accredited(U).

eligible_ok(X) :- bachelor_ok(X), accreditation_ok(X).
```

Deciding Acceptance

```
bachelor_ok(X) :- bachelor(X,is).
bachelor_ok(X) :- bachelor(X,ba).
bachelor_ok(X) :- bachelor(X,it).

accreditation_ok(X) :- university(X, U), accredited(U).

grade_ok(X) :- grade(X,a).
grade_ok(X) :- grade(X,b).

prof_experience_ok(X) :- prof_experience(X,W), W >= 6.

accept_ok(X) :- bachelor_ok(X),
                accreditation_ok(X),
                grade_ok(X)
                prof_experience_ok(X).
```

Deciding Tuition Fee

```
tuition_fee(X, 700) :- citizenship(X, swiss).
tuition_fee(X, 700) :- citizenship(X, eu).
tuition_fee(X, 7500).
```

Solution developed in Class

```
applicant(tim).
studied_in(oxford, other, a, tim).
work(36, tim).
citizenship(eu, tim).

accredited(oxford).

bachelor(ba).
bachelor(is).
bachelor(it).

grade(a).
grade(b).

eligible_ok(X) :- applicant(X),
                  studied_in(U,D,_G,X),
                  accredited(U),
                  not(D = none).

accept_ok(X) :-  applicant(X),
                  studied_in(U,D,G,X),
                  accredited(U),
                  bachelor(D),
                  grade(G),
                  work(W,X),
                  W >= 6.
```