Description

When I started my modeling I have taken a look at the process negotiation pdf uploaded on moodle. The classes Win Condition, Issue, Agreement and Option are modeled similarly.

First of all, I have a class for an Issue. Here I have a unique identifier (ID), a description(STRING) and an issue type(STRING), so that it can be made evident what kind of issue it is (conflict, risk or uncertainty). Here I have a many-to-many relationship with the WinCondition class, as in one win condition there can be multiple issues involved.

The second class is the Option one, where I also have an id, a description and an attribute which shows which issue it addresses.

The third class is the Agreement class. Here I have the first two attributes from above and a list with all the available options to a given agreement. It has also an “adoptedOption” field, which makes it clear which option should be developed in the agreement.

The fourth class is the Win condition where I have and id, a description as above. It also has an optional “refines” attribute, in case it refines another win condition. I also have a list of all the issues it addresses and a list of all the agreements included in the win condition itself. Here I also used my concept of choice : Priority.

I asked one of the other students to send me the “Requirements Negotiation” slides from a lecture from last semester in which I found two relevant field for this class: business importance and ease of realization, both of them hold a value from 0 to 10. Each and every win condition has one priority and a given priority can only be for one win condition.

The first three classes have a superclass: Votable. I decided to do it like this, as Stakeholders can only vote for these.

The Actor and Stakeholder classes are close to each other, the only difference being that Stakeholders have the possibility to vote for “votables”. The Actor class is the parent of the Stakeholder, both of them can be affected by a win condition, issue, option or agreement.

Creating a Vote class was also necessary to make it clear which stakeholder voted for which Votable. Thus, I have created two fields, stakeholderID and votableID, each of them referencing the corresponding classes.

Above all, I created a Generic Element class, which holds an ID and a description, all of the other classes extend this to keep an abstract form.