ASCT 2223 Assignment

Introduction

An important document one must have when entering the job market is a *curriculum vitae* (CV). Actually, it's a document that will accompany each of us during our entire working career and that must always be updated.

There are different types of CVs. For people working in industry it's usually short (maybe just 2 or 3 pages), but for academics it's much more comprehensive with dozens of pages. It may be text oriented or filled with images showing works (like decor or drawings or even artistic pieces).

There are several forms of making it available, either through a PDF, a web page or even as a video. In some cases, the same CV may need to be available in several formats, making its updating time consuming.

Part I

What To Do

In this first part of the assignment you will create a metamodel for CVs. It should be generic enough to accommodate as many kinds of CVs as possible. For instance, you should consider sections from job positions, to skills, to bibliography entries (e.g. books, scientific articles), to art pieces, etc. To achieve this final goal, you should follow these two steps[1]:

1. Modeling Domain Analysis

As a first step, you should start by finding several examples of concrete CVs (including yours) so you can better understand the domain. Several resources exist online; in particular www.cienciavitae.pt is the national science foundation CV platform where you can find CV for researchers of all areas (do you know anything similar for other kinds of CVs?). If you gather several CVs, it will be easier to better understand the domain and define the necessary concepts to include in your metamodel. From this task, you should create a list of concepts that must be part of the language you are creating (e.g. header, job title, etc.). For each concept, you should also define its attributes (e.g. a bibliography entry may have a title, authors, and so on). Finally, each concept may be connected to other concepts (e.g. the section "personal information" may be connected to the address). For an example you may consult Figure 7.4 of the book [1]. Note this list of concepts is a deliverable. I suggest you create a table similar to the one presented in the book.

[1] More about this in section 7.3 of [1].

2. Modeling Language Design

After defining the concepts of your language, it's time to create a metamodel with them. This should be guided by the information from the previous step. This should be done using the EMF framework (if you intend to use others, please let me know). In this step you should also take under consideration that not all constraints are possible to be defined in the metamodel. Thus, consider the use of OCL to make your metamodel closer to reality adding the necessary constraints. Note the metamodel (and OCL) is also a deliverable.

Deliverables

You should produce at least two deliverables for this part I of the assignment:

1. A list of concepts of the language (including its attributes and connections).
2. A metamodel of the language (possibly including OCL constraints).

The deadline for a first version of these deliverables is October 28th, midnight. You may change these deliverables in the following phases of the assignment, but it's important to have a base in the following weeks.

You should form a group of 3 people, create a repository to save your deliverable and share it with me (jacomecunha @ github, for instance).

References

[1] Model-Driven Software Engineering in Practice: Second Edition 2nd Edition by Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan & Claypool Publishers; 2nd edition (March 30, 2017), ISBN-13: 978-1627057080.