Université Catholique de Louvain

LINGI2172 - Databases

Mission 3 - Database Design

Auteur :
Baugnies Benjamin (6020-10-00)
Colson Olivier (5039-10-00)
Vanwelde Romain (3143-10-00)

Superviseurs :
Pr. Bernard Lambeau
Antoine Cailliau

3 avril 2014

1 Introduction

Telling a story is challenging. Indeed, in order to build a “good" scenario, one must think of a lot of different aspects and ensure coherence between all these points.

Some can manage it using diagrams, other can rely on tons of paper sheets referencing each other or, more reasonably, use a computer to store all their documents. However, with all these ways of working, the same problem arises : as the storyline and background get denser, it becomes more and more difficult to ensure that no contradiction appears. This is a big problem, since contradictions ruin the feeling of reality that must always be given by a good scenario. How about asking the computer to gather, interpret and display all this information in a clean and understandable way ?

Our project can be defined as a “narration manager". Its goal is to make it easier for people to write coherent and complex scenarios without either becoming mad or cancelling their project because of its increasing complexity. It is intended for all "story makers" (videogame makers, film makers, roleplayers, writers, . . .), and is thus meant to be generic and conveniently adapt to various situations, as well as user preferences and priorities in the story (for example, some users could want to define a precise date for each event happening in their story, as others could prefer to focus on the relations binding all the characters together).

Possibilities of telling a story are infinite, yet time and coordination constraints often limit what is actually possible to achieve. It is now time to push these limits away.

2 ORM schema

The ORM schema is shown on the Figure 1. If some relations are too difficult to read, the numeric version is available in the Annexes directory of the zip file.

As we can see on the schema, there are 5 main entities which are "`Characters`", "`Date`", "`Events`", "`Place`", and "`Map`".

We will explain here above three specific case of the diagram :

Character - Relation relations involving time range, time or timeless notion.

The relations explain the relationships between different characters. They are uni-directional and the different kinds of relations can be defined by the user. We split these relations into three types to represent the fact that some relations are time-independent (e.g ". . . is my father"), some can start at a given time and be permanent and/or open-ended (e.g ". . . is my godfather since . . . "), and finally some can last for only a while (e.g ". . . was my friend from . . . to . . . ").

Pseudo relation.

This relation involves two characters and a pseudonym. It describes the name used by the first character for the second one. This represents the fact that during the story, a given character might not know another's real name. We added this relation since this can have an impact on the story (he wouldn't realize others were talking about someone he knows for example). The corresponding table will also allow us the find all the pseudonyms a given person might go by.

Place - Map relation.
This is a rather complex relation that we introduced to keep track of a story's geography at different levels. The first use is to allow users to situate events or characters. We can also define sub-places to refine locations. We can see that a place's map is optional. However, since a place's sub-places are linked to its map, it implicitly becomes required when we want to add levels. This structure allows us to chain an arbitrary number of levels with a "place - map - place - map - . . . " hierarchy. This relation has two constraints that are not expressed in the database and will have to be verified in the software implementation. Firstly, the map square on which the sub-place is located must belong to the map's domain of possible squares (between 1 and (# square width)*(# square length)). Secondly, it is required that two places of the same level do not overlap.

3 Relvar predicates

3.1 About characters

Pierre is from Bruxelles.
Pierre is born on 28/12/1992
Jean is born on 01/04/1992
Benjamin is born on 03/06/1992
Jean is melancholy
Benjamin is member of the association "Les Petits Riens"

3.2 Characters relations

Pierre likes Benjamin
Benjamin likes Pierre
Benjamin likes Jean
Jean doesn't like Benjamin
Paul is Pierre's father

3.3 About events

Jean attended the event "The beer festival"
The "beer festival" took place at LLN
The "beer festival" took place on 08/03/2014
The "beer festival" is "blablablablablablablabla" as description.

3.4 About places

Intel room is a sub-location of Réaumur
Réaumur is a sub-location of LLN
The map "LLNMap" represents the location "LLN"
The map "RéaumurMap" represents the location "Réaumur"
Réaumur is represented on the LLNMap at square number 5.
LLNMap has 10 square width, and represents a 5km distance.
LLNMAP has 20 square length, and represents a 10km distance.

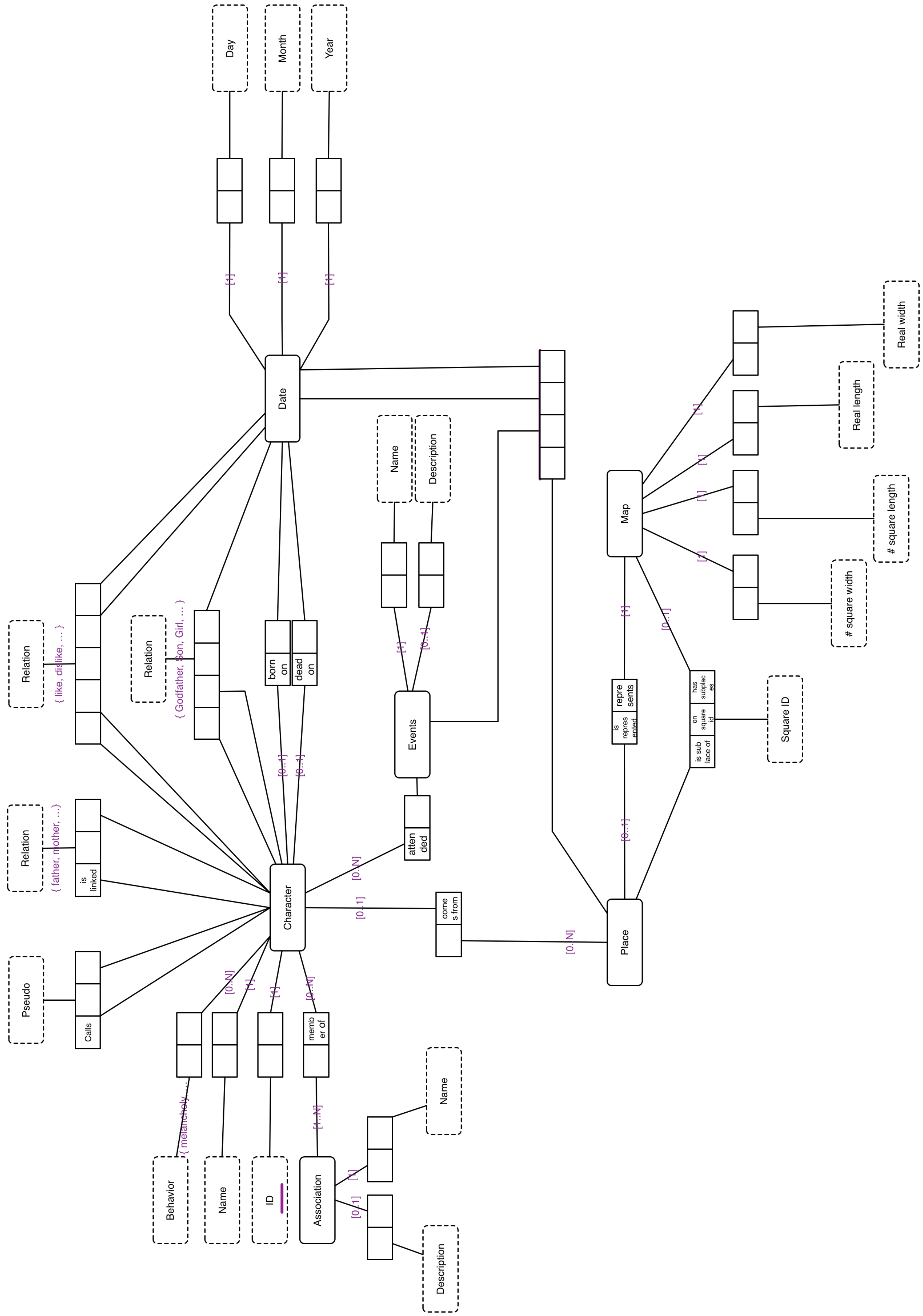

Figure 1 – ORM Schema