Dagstuhl Relationships and/or Roles plus Queries Beauty Contest

proposed by Friedrich Steimann

I: Problem

Data

A play consists of acts, which are divided into scenes, which are further divided into speeches. Each speech is assigned to a role. A play is performed by an ensemble. An ensemble is comprised of actors, each of whom plays a single role of the performed play.

Functions

- To list the acts an actor has an appearance in.
- To print a role script for a given role, listing each speech assigned to the role, surrounded by the immediately preceding and immediately following speech, if any (assigned to other roles; no repeats of speeches), from the same act. For example, the role script for the role *Second Witch* of *The Tragedy of Macbeth* begins with

  > *First Witch*: When shall we three meet again/In thunder, lightning, or in rain?
  >
  > *Second Witch*: When the hurlyburly's done,/When the battle's lost and won.
  >
  > *Third Witch*: That will be ere the set of sun.
  >
  > *First Witch*: Where the place?
  >
  > *Second Witch*: Upon the heath.
  >
  > *Third Witch*: There to meet with Macbeth.

The Problem V2

Evolve the program so that a speech can be assigned to many roles ("chorus"), an actor can play several roles ("double roles"), and that a performance can be associated with a series of dates, some of which may be classified as rehearsals.

II: Hints for Solutions

- The solution should reveal a data structure relying on relationships and/or roles as programming primitives.
- The solution may contain expressions in query languages (such as JQL or LINQ).

III: Challenges and Evaluation Criteria

- How are binary relationships represented?
- How does changing a cardinality (or multiplicity) from one to many affect the program?
- How are derived relationships (e.g., from an actor to an act) handled or supported?
- How are ternary relationships (e.g., performance of a play by an ensemble at a date) represented?
- The problem contains several examples of bidirectional relationships, whose direct representation can however be avoided (see attached sample solution). How does changing the directionality of a relationship from unidirectional to bidirectional affect the program in terms of cost and benefit?
- Can the limited recursive composition structure (a play consisting of acts, acts consisting of scenes, scenes consisting of speeches) be represented and/or exploited? Does the approach offer any savings when compared to representing each composition as a separate relationship? Can a possible composition relationship defined for plays be reused for ensembles composed of actors?
- In the example, *role* may be both a domain and a programming language concept. How does implementing roles with roles translate to the metalevel (language specification)?