CSC343 A3

ER MODELLING AND DATABASE DESIGN

Ross Gatih 997 92 311 8
Trong Truong 995 94 222 6

Sunday 18 November 2012

Contents

Part I

Assumptions

1 Setup and login phase

Number of reviewers. Each conference has a fixed number of reviewers assigned per paper.

Conference participant multiplicity. Every user can participate in at most one role per conference, otherwise we have a conflict of interest. (E.g. a chair has the power to choose who gets to review a paper, so she might choose herself.)

Conference topic multiplicity. Every conference can define its own topics, or use existing topics from other conferences.

File storage. We'll only store filenames of papers, forms, and letters in the database. The files themselves will be stored on a hard drive somewhere outside the database.

2 Submission phase

Author conferencce participation. An author may have an account without being registered in any conference. She may upload a paper without submitting it to any one.

Different submissions are different papers. An author has the option to update an existing submission instead of submitting an identical copy, but that's application, not database-level design, so we don't have to worry

about that.

Author VS coauthor. We don't distinguish between authors and coauthors: everyone author of a paper is a coauthor.

3 Reviewing phase

Reviewer preference: bidding process. To bid on a paper, a reviewer expresses her level of interest by assigning an integer between 0 (low) and 5 (high).

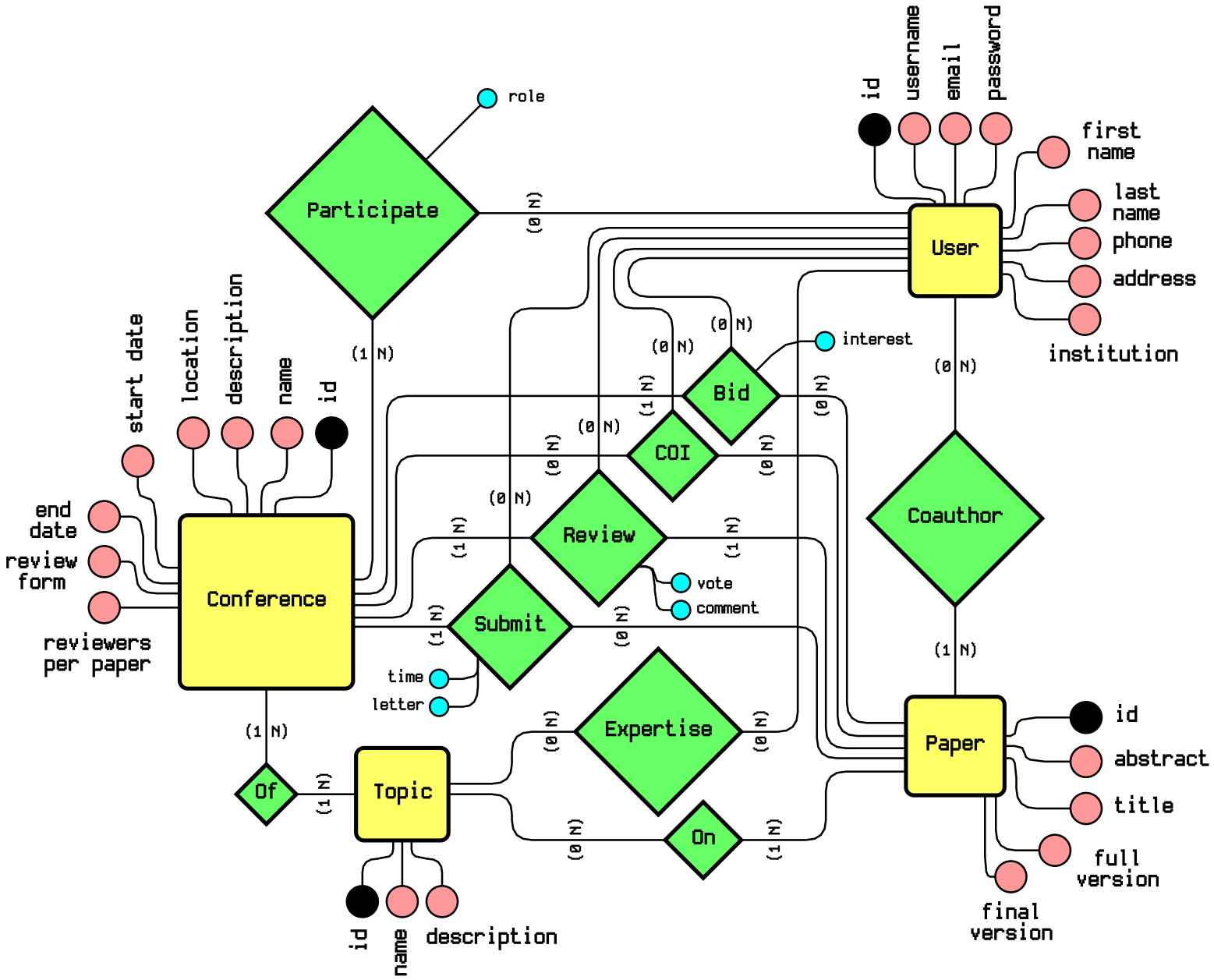
Reviewer's choice. A reviewer casts a vote of YES or NO on whether or not she thinks the paper should be published.

4 Decision and publishing phase

One letter per paper. The chair sends only one acceptance notification letter per paper, to the coauthor who submitted it.

Part II

ER diagram


role
Participate
(0 N)
User
id
username
email
password
first name
last name
phone
address
institution
(1 N)
start date
location
description
name
id
(0 N)
interest
Bid
(1 N)
(0 N)
COI
(0 N)
Coauthor
end date
review form
reviewers per paper
Conference
Review
vote
comment
Submit
time
letter
Expertise
Paper
id
abstract
title
full version
final version
Of
Topic
On
id
name
description

Part III

Relational schema

Users (ID, Username, E-mail, Password, First name, Last name, Institution, Address, Phone number)

Expertise (userID, Topic ID)

Bid (userID, Paper ID, Conference ID, Interest)

COI (userID, Paper ID, Conference ID)

Review (userID, Paper ID, Conference ID, Vote, Comment)

Submission (userID, Paper ID, Conference ID, Time, Letter)

Paper (Paper ID, Abstract, Title, Full version, Final version)

Author (Paper ID, userID)

PaperTopic (Paper ID, Topic ID)

Conference (Conference ID, Name, Description, Location, Start date, End date, Review form, Reviewers per paper)

members (Conference ID, userID, Role)

ConferenceTopic (Conference ID, Topic ID)

Topic (Topic ID, Name, Description)

Part IV

PostgreSQL database definition

```
CREATE TABLE Conference
```

```
(
    cid                 SERIAL NOT NULL PRIMARY KEY,
    name                VARCHAR (255) NOT NULL,
    description         VARCHAR (255) NOT NULL,
    location            VARCHAR (255),
    start_date          DATE,
    end_date            DATE,
    review_form         VARCHAR (255), -- Link to file
    num_reviewer        INT
);

-- users because user is a reserved word in psql
CREATE TABLE Users
(
    uid                 INT NOT NULL PRIMARY KEY,
    username            VARCHAR(255) UNIQUE NOT NULL,
    email               VARCHAR(255) UNIQUE NOT NULL,
    password            VARCHAR(255) NOT NULL,
    first_name          VARCHAR(255),
    last_name           VARCHAR(255),
    institution         VARCHAR(255),
    address             VARCHAR(255),
    phone               VARCHAR(255)
);

CREATE TABLE Paper
(
    pid                 INT NOT NULL PRIMARY KEY,
    abstract            TEXT NOT NULL,
    title               TEXT NOT NULL,
    full_version        VARCHAR(255), -- Links
    final_version       VARCHAR(255)
);

CREATE TABLE Topic
(
    tid                 INT NOT NULL PRIMARY KEY,
    name                VARCHAR(255) NOT NULL,
```

```
    description         TEXT
);

CREATE TABLE Expertise
(
    uid                 INT NOT NULL REFERENCES Users (uid),
    tid                 INT NOT NULL REFERENCES Topic (tid),
    PRIMARY KEY (uid, tid)
);

CREATE TABLE Bid
(
    uid                 INT NOT NULL REFERENCES users(uid),
    pid                 INT NOT NULL REFERENCES paper(pid),
    cid                 INT NOT NULL REFERENCES conference(cid),
    PRIMARY KEY (uid, pid, cid)
);

CREATE TABLE Coi
(
    uid                 INT NOT NULL REFERENCES Users (uid),
    pid                 INT NOT NULL REFERENCES Paper (pid),
    cid                 INT NOT NULL REFERENCES Conference (cid),
    PRIMARY KEY (uid, pid, cid)
);

CREATE TABLE Review
(
    uid                 INT NOT NULL REFERENCES users(uid),
    pid                 INT NOT NULL REFERENCES paper(pid),
    cid                 INT NOT NULL REFERENCES conference(cid),
    vote                INT CHECK (0 <= vote AND vote <= 5),
    comment             TEXT,
    PRIMARY KEY (uid, pid, cid)
);

CREATE TABLE Submission
(
```

```
    uid                 INT NOT NULL REFERENCES Users (uid),
    pid                 INT NOT NULL REFERENCES Paper (pid),
    cid                 INT NOT NULL REFERENCES Conference (cid),
    time                TIME NOT NULL,
    letter              VARCHAR (255) NOT NULL,
    PRIMARY KEY (uid, pid, cid)
);

CREATE TABLE Author
(
    uid                 INT NOT NULL REFERENCES Users (uid),
    pid                 INT NOT NULL REFERENCES Paper (pid),
    PRIMARY KEY (uid, pid)
);

CREATE TABLE PaperTopic
 (
    pid                 INT NOT NULL REFERENCES paper(pid),
    tid                 INT NOT NULL REFERENCES topic(tid),
    PRIMARY KEY (pid, tid)
);

CREATE TABLE Members
(
    cid                 INT NOT NULL REFERENCES conference(cid),
    uid                 INT NOT NULL REFERENCES users(uid),
    PRIMARY KEY (cid, uid)
);

CREATE TABLE ConferenceTopic
(
    cid                 INT NOT NULL REFERENCES Conference (cid),
    tid                 INT NOT NULL REFERENCES Topic (tid),
    PRIMARY KEY (cid, tid)
);
```