Database Entity-Relationship Model

Revision History

Revision 1.00 November 17, 2003 Revised by: afw
Initial version.
Revision 1.01 August 13, 2004 Revised by: afw
Added additional entities.
Revision 1.02 May 9, 2006 Revised by: jd
Added entitie team.
Revision 1.03 June 27, 2008 Revised by: jd
Added entity folder and folder_id in job.

An entity relationship model for the Spk database is presented.

Table of Contents

Introduction

In Spk, job status, job history, and user data are maintained in a relational database. The database is the sole means of communication between the major components of the server side of the application: namely, the web server, the SPK compiler, and the computational server. This document presents the ER model for this database.

Entity Relationship Diagram

The following diagram represents the SPK database in terms of data entities and relationships. In the following diagram, note the following:

- Entities are represented by boxes that are divided into sections, vertically. The name of the entity appears in the upper section. A list of the most important attributes appears in the lower section.
- Relationships between entities are represented by connector lines.
  - Each relationship is modeled as a *has-a* relationship. For example, a user *has* zero or more jobs. A job *has* a state.
  - An arrow-head at one end of a connector line points to the entity which *has*. In the example of the line that connects user and job, the arrow-head points toward user, because it is the user who has jobs and not the other way around.
  - The *multiplicity* of the relationship is represented by the end of the connector opposite the arrow-head. For example, the "crow's-foot" symbol at the job end of the connector between user and job implies that the user may have multiple jobs. The lack of a crow's-foot in the connector between job and model implies that a job can have at most one model.

The SPK database is implemented as a relational database. The high level design is documented in the Database Schema [1] specification. The entity-relationship model

and the database schema are complementary and non-redundant. As implemented in SPK, there is a one-to-one relationship between entities and database tables as well as a one-to-one relationship between entity attributes and table columns. The connector lines in the er-diagram clearly indicate the direction and multiplicity of relationships. This information is difficult or impossible to infer from the schema alone.

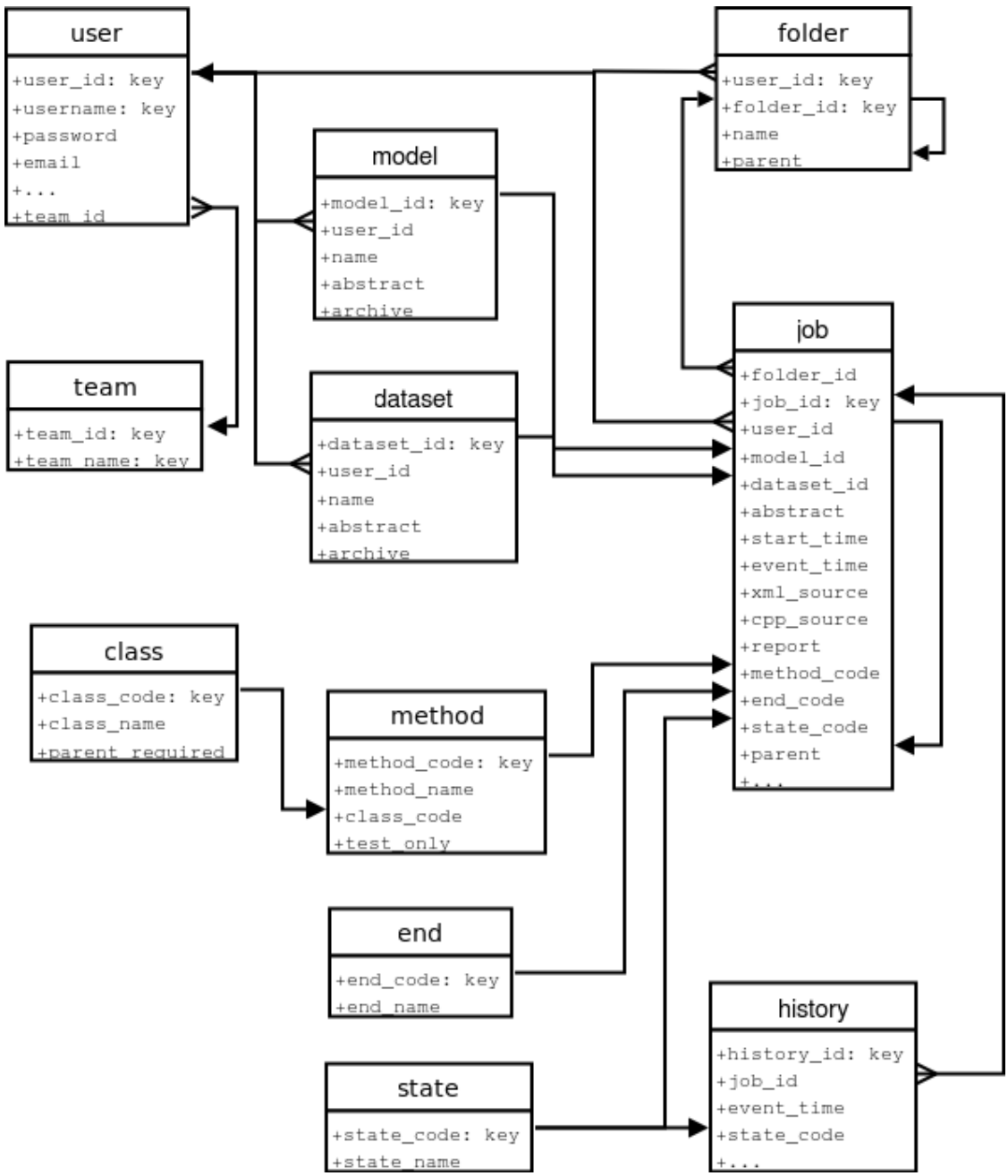

Figure 1. Entity Relationship Diagram

Entities

In this section, each entity will be described briefly.

Job

Job represents the principal unit of work for Spk. A user combines a model and a dataset to create a job. A user may have many jobs. Some jobs contain a non-zero parent link, which associates the job to another job that was previously completed.

Folder

Each instance of folder entity represents a job folder. A job folder belongs to a user. Folder has name and its parent folder.

History

Each instance of a history entity represents one of the states assumed by a job. The date and time of transition into that state is recorded. For more information, see the JobHistory Model [2] specification.

Model

Each instance of the model entity represents a PK model created by a user with the aid of the MDA. The XML source for the model is contained in the *archive* attribute as an rcs-compatible version controlled archive.

Dataset

Each instance of the dataset entity represents a set of data created or imported by a user with the aid of the MDA. The XML source for the dataset is contained in the *archive* attribute as an rcs-compatible version controlled archive.

User

User represents the user of Spk. A user may have many jobs, many models and many datasets.

Team

Team represents the user group of Spk. A team may have many users, but a user can only join one team.

Method

Spk supports various computational methods. Each job uses one such method. Certain methods are available only to users who have "test" privileges.

Class

Each method belongs to a method class. Some methods can only be applied to a job which has a non-zero parent attribute linking it to a previously completed job.

State

At any given moment, a job is in exactly one state, which must relate to an instance of the state entity. Each instance of the state entity contains a state_code and a longer state_name, which is English text. In the future, there may be versions of SPK that use languages other than English. In that case, a name attribute will be added for each additional language supported.

End

When a job reaches the 'end' state, its end attribute is non-null and indicates the nature of its completion. Each instance of the end entity contains an end-code and a longer end_name, which is English text. In the future, there may be versions of SPK that use languages other than English. In that case, a name attribute will be added for each additional language supported.

Notes

1. ../dbSchema/dbSchema.html
2. ../jobHistory/jobHistory.html