1 Spicey: An ER-based Web Framework

Spicey is a framework to support the implementation of web-based systems in Curry. Spicey generates an initial implementation from an entity-relationship (ER) description of the underlying data. The generated implementation contains operations to create and manipulate entities of the data model, supports authentication, authorization, session handling, and the composition of individual operations to user processes. Furthermore, the implementation ensures the consistency of the database w.r.t. the data dependencies specified in the ER model, i.e., updates initiated by the user cannot lead to an inconsistent state of the database.

1.1 Installation

The actual implementation of Spicey is a package managed by the Curry Package Manager CPM. Thus, to install the newest version of Spicey, use the following commands:

```
> cypm update
> cypm install spicey
```

This downloads the newest package, compiles it, and places the executable `spiceup` into the directory `$HOME/.cpm/bin`. Hence it is recommended to add this directory to your path in order to execute Spicey as described below.

1.2 Basic usage

The idea of this tool, which is part of the distribution of Curry, is described in detail in [2]. Thus, we summarize only the basic steps to use this tool in order to generate a web application.

First, one has to create a textual description of the entity-relationship model in a Curry program file as an (exported!) top-level operation type `ERD` (w.r.t. the type definitions defined in the module `Database.ERD` of the package `cdbi`) and store it in some program file, e.g., "`MyERD.curry`". The directory `examples` in the package `spicey`[1] contains two examples for such ERD program files:

`BlogERD.curry`: This is a simple ER model for a blog with entries, comments, and tags, as presented in the paper [2].

`UniERD.curry`: This is an ER model for university lectures as presented in the paper [1].

Then you can generate the sources of your web application by the command

```
> spiceup MyERD.curry
```

with the ERD program as a parameter. You can also provide a file name for the SQLite3 database used by the application generated by Spicey, e.g.,

```
> spiceup --db MyData.db MyERD.curry
```

If the parameter "`--db DBFILE`" is not provided, then `DBDFILE` is set to the default name "*ERD*`.db`" (where *ERD* is the name of the specified ER model). Since this specification will be used in the *generated* web programs, a relative database file name will be relative to the place where the web

[1] If you installed Spicey as described above, the downloaded `spicey` package is located in the directory `$HOME/.cpm/app_packages/spicey`.

programs are stored. In order to avoid such confusion, it might be better to specify an absolute path name for the database file. This path could also be set in the definition of the constant `sqliteDBFile` in the generated Curry program `Model/`*ERD*`.curry`.

Spicey generates the web application as a Curry package in a new directory. Thus, change into this directory (e.g., `cd` *ERD*) and install all required packages by the command

```
> make install
```

The generated file `README.txt` contains some information about the generated project structure. One can compile the generated programs by

```
> make compile
```

In order to generate the executable web application, configure the generated `Makefile` by adapting the variable `WEBSERVERDIR` to the location where the compiled cgi programs should be stored, and run

```
> make deploy
```

After the successful compilation and deployment of all files, the application is executable in a web browser by selecting the URL `<URL of web dir>/spicey.cgi`.

1.3 Further remarks

The application generated by Spicey is a schematic initial implementation. It provides an appropriate basic programming structure but it can be extended in various ways. In particular, one can also use embedded SQL statements (see [3] for details) when further developing the Curry code, since the underlying database access operations are generated with the `cdbi` package. The syntax and use of such embedded SQL statements is sketched in [3] and described in the Curry preprocessor.

References

[1] B. Braßel, M. Hanus, and M. Müller. High-level database programming in Curry. In *Proc. of the Tenth International Symposium on Practical Aspects of Declarative Languages (PADL'08)*, pages 316–332. Springer LNCS 4902, 2008.

[2] M. Hanus and S. Koschnicke. An ER-based framework for declarative web programming. *Theory and Practice of Logic Programming*, 14(3):269–291, 2014.

[3] M. Hanus and J. Krone. A typeful integration of SQL into Curry. In *Proceedings of the 24th International Workshop on Functional and (Constraint) Logic Programming*, volume 234 of *Electronic Proceedings in Theoretical Computer Science*, pages 104–119. Open Publishing Association, 2017.