

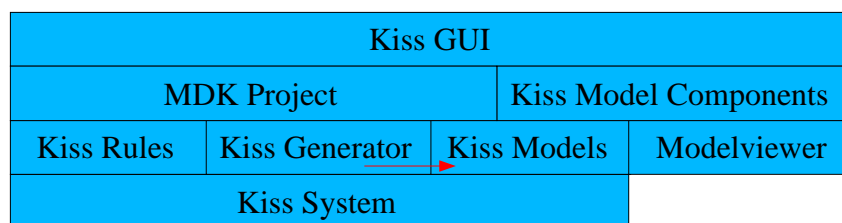
Author: J.F. Benckhuijsen (jfbenck@users.sourceforge.net)

Version: 1.0

MDK Overall design

1. Overall design

The MDK program consists out of several libraries, which interact (and thus depend) in some way or another. Each of these libraries have their own responsibilities. I'll first show a graphical view of the libraries and the dependencies that exist between them. After that, I'll explain the responsibilities of each of these libraries.



The red arrow from Kiss Generator to Kiss Models means that the Kiss Generator Library uses parts of the Kiss Models Library. This is to prevent duplication of code.

Kiss System Library

MDK is used to create an application. Each application is created as an equivalent of a certain real life system. The Kiss System Library stores all characteristics of a system. The design of the system finds its origins in the Kiss Modeling Language (<http://www.kiss.nl>) on which this program is based.

Kiss Rules Library

The Kiss System library is only a storage library. It doesn't do any checking to ensure the consistency of the system as a whole, only technical checks are implemented here. It provides however a framework to perform these checks. The Kiss Rules Library stores all logical checks a system has to obey to according to the Kiss Modeling Language. It provides an extendible way to store these rules, without creating a dependency between otherwise independent parts of the system.

Kiss Models Library

The Kiss System library describes a whole system. The Kiss Models library stores the classes which translates all information about the system to various models supported by the Kiss Modeling Language. It hides the complexity of the models from the User Interface. It also provides an interface to the User Interface to edit the models. The Kiss Models library translates these changes to changes to the Kiss System.

Kiss Generator Library

The main purpose of MDK is to generate an application using the information from the models. The Kiss Generator library offers this. It translates the information stored in the Kiss System into a (framework of a) web application which runs on PHP/MySQL.

Modelviewer Component

The Modelviewer component is a generic component to draw and edit different kind of models. Various models offered by MDK are based on this component. These models use the functionality offered by the modelviewer to implement their own functionality.

MDK Project Library

The MDK Project library manages a complete project created with MDK. It allows models to be created, saving and loading of the projects, generation of the web application and handles all technical aspects like memory allocation.

Kiss Models Components

The Kiss Models Components is a set of components which are used to build the User Interface. They offer the representation part of the models and are responsible to draw the different models and to translate the input by the user to calls to the Kiss Models Library.

These components are implemented as cross-platform as possible. However, as they are part of the User Interface, they are somewhat platform specific. Our main development platform (for now) is MS Windows / Borland C++ Builder 5.

Kiss GUI

The Kiss GUI is the Borland C++ Builder 5 implementation of the GUI of the MDK program. The largest part consists out of a framework to show the different models.