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

Version: 1.0

Kiss System Library

1. General design

The Kiss System library is offers a set of classes to store all information about a system according to the Kiss Modelling Language. The UML-diagram of these classes is shown below. For clarity, i've removed some of the base classes. They are shown in the inheritence diagram below.

The main structure of the Kiss System Library reflects the way Kiss sees a system. A system generally exsists out of Subjects, Actions, Processes and Objects. These four classes and their associations are the basis of the whole model. The rest of the classes are just extensions to the model, to store more information.

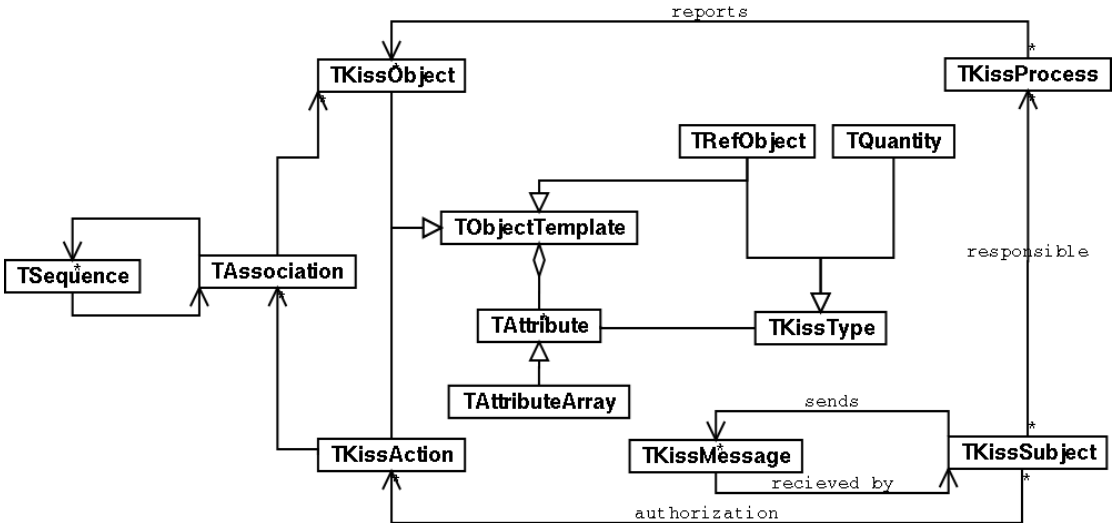


Figure 1 : General overview

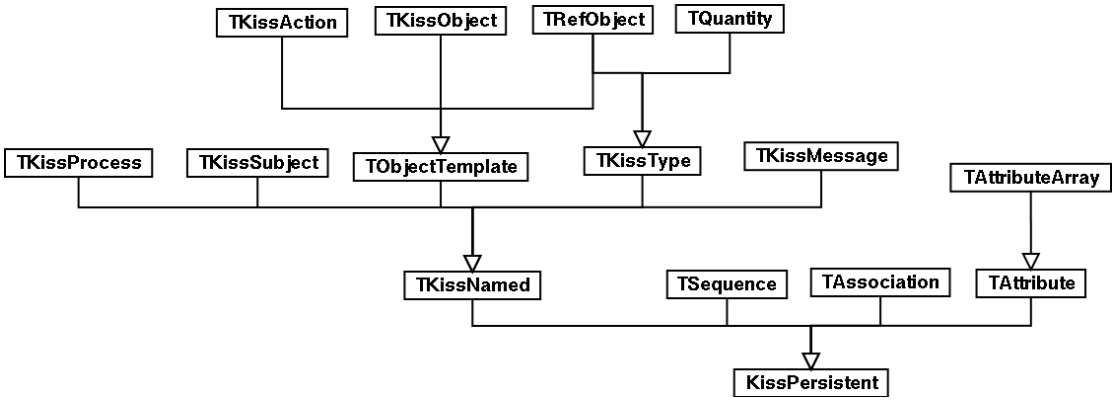


Figure 2 : Inheritence

2. Individual classes

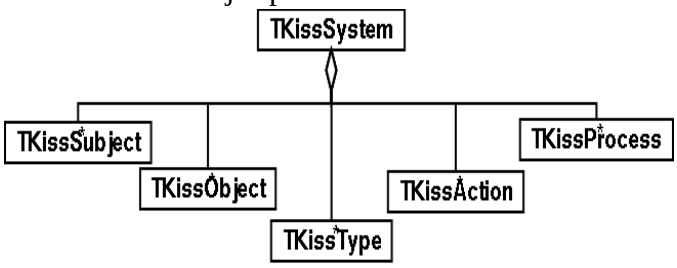
KissPersistent

This is the base class of all classes in the kiss system. It offers to abstract

methods load() and save(). These are used to load and save all the data of that class.

KissSystem

The KissSystem class is a large container class. It stores and manages all data in the system. In Kiss a system consists out of 5 major parts:



1. Subjects -> The actors of a system
2. Actions -> The actions excecuted by the actions
3. Processes -> The processes subjects are responsible for
4. Objects -> The objects the actions modify and processes report about
5. Types -> The types of the attributes of actions and objects

The KissSystem class is a container for each of these 5 major parts.

KissNamed

Kiss depends heavily on named objects. Just about everything of importance within the system has a name. This is the base class of everything that has a name in the Kiss System

KissSubject, KissMessage, KissProcess

This stores the information about a subject (or in UML-talk an Actor), which is important to the system. Subjects send and receive messages, can be responsible for certain processes within the system and are authorized to excecute certain actions.

ObjectTemplate, KissType, Quantity and RefObject, Attribute, AttributeArray

Objects and actions both have attributes associated with them. ObjectTemplate is a generalization of both to store these attributes. An attribute has a certain type : KissType. This itself is an generalization of Quantity and RefObject. An quantity is an atomic type, like length, time, names, etc. Reference Objects are types which consist out of other types. An example is the personal data of a person. This consists out of a name, an address, zip code, etc.

AttributeArray support associative arrays, where the index is not simply an integer, but for example a name or a timestamp.

KissAction, KissObject, Association, Sequence

In Kiss subject excecute actions on object they are associated with. These associations have additional information about what kind of association is meant, and what logics will be excecuted when an action is preformed. Also actions can only be excecuted in a certain order, a certain sequence. For example someone can't deposit money to an account before it's opened. The sequences

store the way of life of an object.

2.2 Additional modules

KissSignal

The Kiss System library uses the LibSigC++ Library to support signals which are emitted as a result of changes in the system. They are implemented to support the Kiss Rules Library (see documentation Rules Library). The Kiss Signal module offers some convienice functions and typedefs for the implementation of the framework of the rules within the Kiss System library

KissXMLNode

The KissXMLNode is used to load and save all the classes within the KissSystem. More information can be found in the IO documentation.