Observation system

Table of contents

1. What is observation ?

By observation we want to express the idea that some objects are able to change automatically their state depending on other objects states (at least this is my understanding and my way to express it). For instancesuppose we have a VertexSet N and an EdgeSet E based upon it. Some user may wish to remove a Vertex v from N. We would then expect that E doesn't contain any edge related to v anymore, without any intervention from the user. This is exactly what happens with Mascopt, and that behaviour is based on what we call "observation".

2. The general principles

Broadly the observation mechanism is implemented in a very simple way. The observed object must extends the ObservableObject class (which mascoptObject extends) and call the right method (generally called notify(Add/Remove/Value)Observers) when something happens. Besides the observer object must implement the Observer interface, that is mainly implements an update method. Moreover, the observer object is responsible for its adding and removing from the the list of observers of the related object. This is done by calling the appropriate methods of the observed object. Those methods are usually named add(Add/Remove/Value)Observer and remove(Add/Remove/Value)Observer.