



Assume that the game is being controlled by a window containing a button to run the clash and an output text field. The code that should be run upon clicking the button can be attached to it.

Does the game with ogres and knights have some user interface?

Is it sufficient to attach to the button the clash code we already have?

WIMP  
windows, icons, menus, pointer  
Xerox PARC

Entangling the user interface with application logic (an inner program model) makes problems in changing the user interface

The inner program model must not be put into listeners

A user interface has to be decoupled from the application logic

The interaction with GUI is based on capturing and handling events

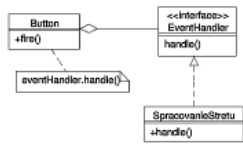
Observer is appropriate for interconnecting the user interface and application logic

Events are handled by corresponding handlers in the context of the user interface elements

## Strategy



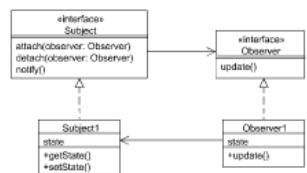
there are (and can be added) different strategies of solving a given problem, and the context of their application has to be ready for this



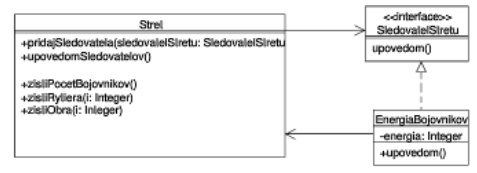
## Observer

A user interface (or its parts) has to be notified of a change in the model, whereas it should be possible to add it without the need to change the model.

observing objects have to be notified of a change in the state of the observed object, and it has to be possible to add them without a need to modify the observed object



A clash is an object. Its state changes. GUI elements could be reacting differently to the change in the state of the clash, e.g., show the current overall energy. How to provide for adding such GUI elements without the need to change the clash object?



Lecture 4:

# Graphical User Interface and Its Separation from Application Logic

Valentino Vranić

Ústav informatiky, informačných  
systémov a softvérového inžinierstva



[vranic@stuba.sk](mailto:vranic@stuba.sk)

[fiit.sk/~vranic](http://fiit.sk/~vranic)

OOP 2018/19

5. 3. 2019

Does the game with  
ogres and knights have  
some user interface?

Assume that the game is being controlled by a window containing a button to run the clash and an output text field.

The code that should be run upon clicking the button can be attached to it.

Assume that the game is being controlled by a window containing a button to run the clash and an output text field.

The code that should be run upon clicking the button can be attached to it.

Is it sufficient to attach to the button the clash code we already have ?

A user interface has  
to be decoupled  
from the  
application logic

WIMP

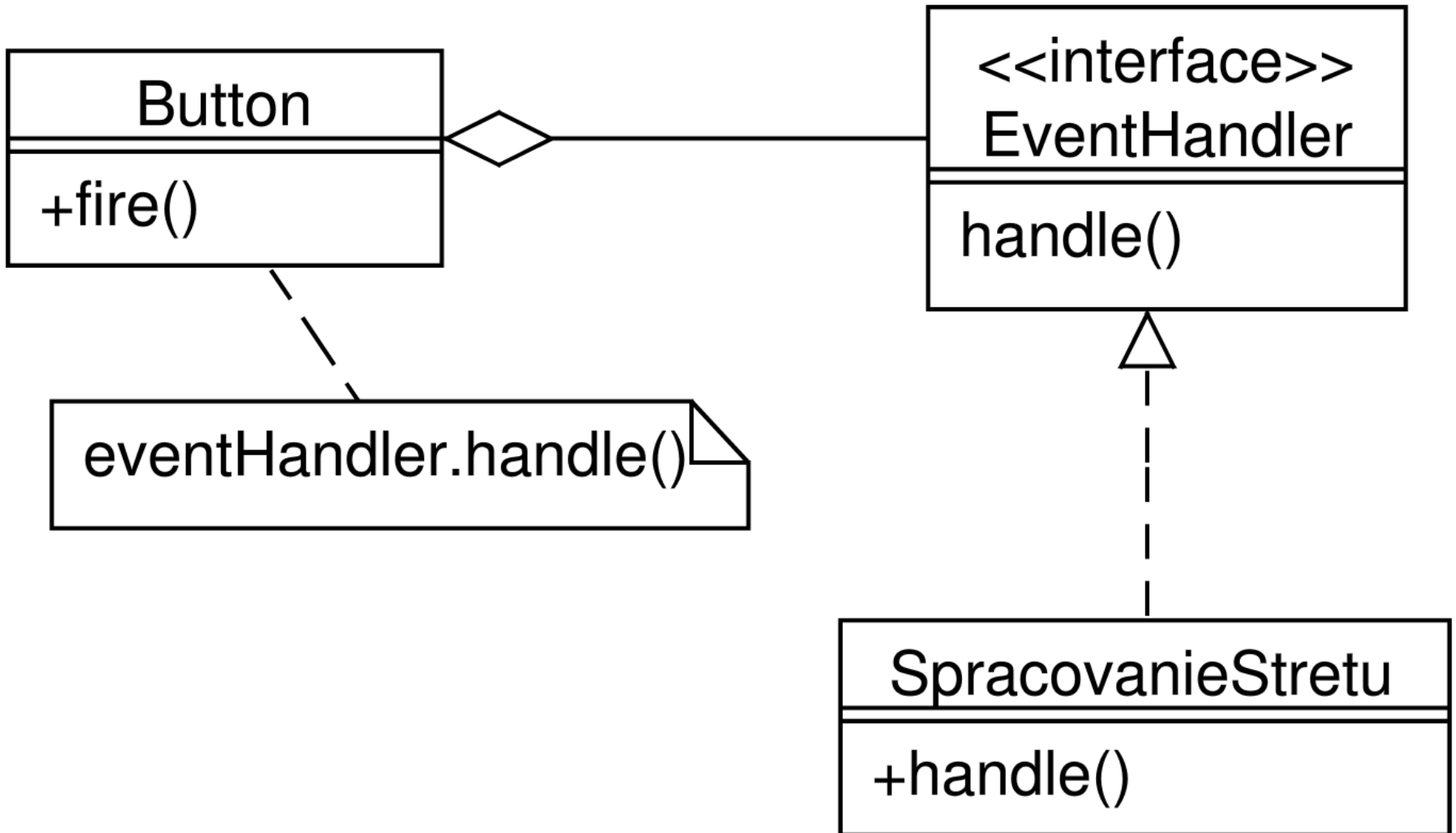
windows, icons, menus, pointer

Xerox PARC

Entangling the user interface with application logic (an inner program model) makes problems in changing the user interface



The inner program model  
must not be put into listeners

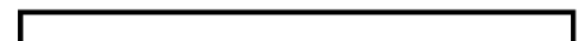
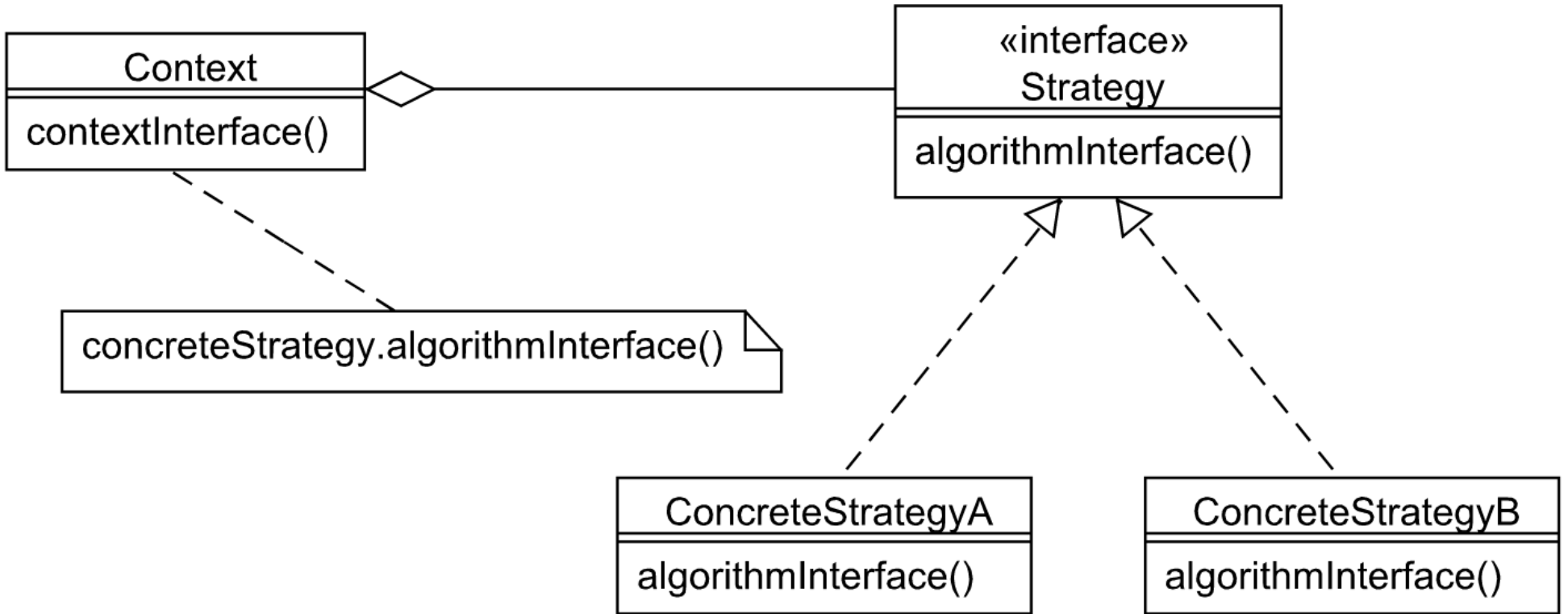


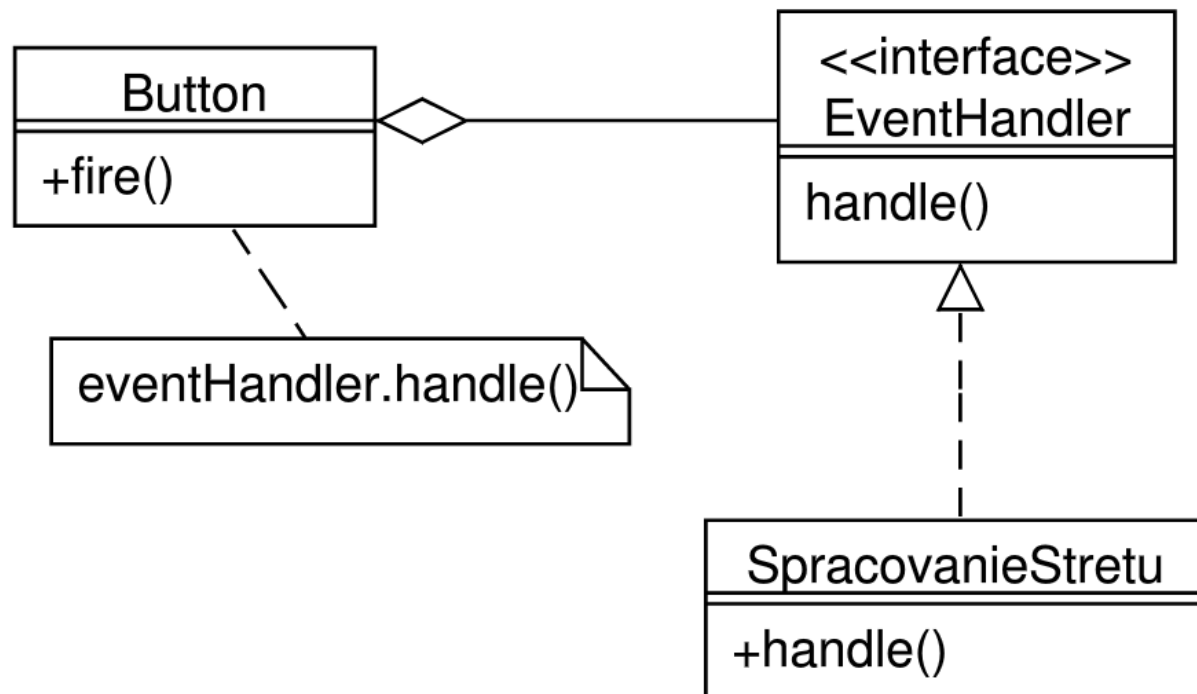
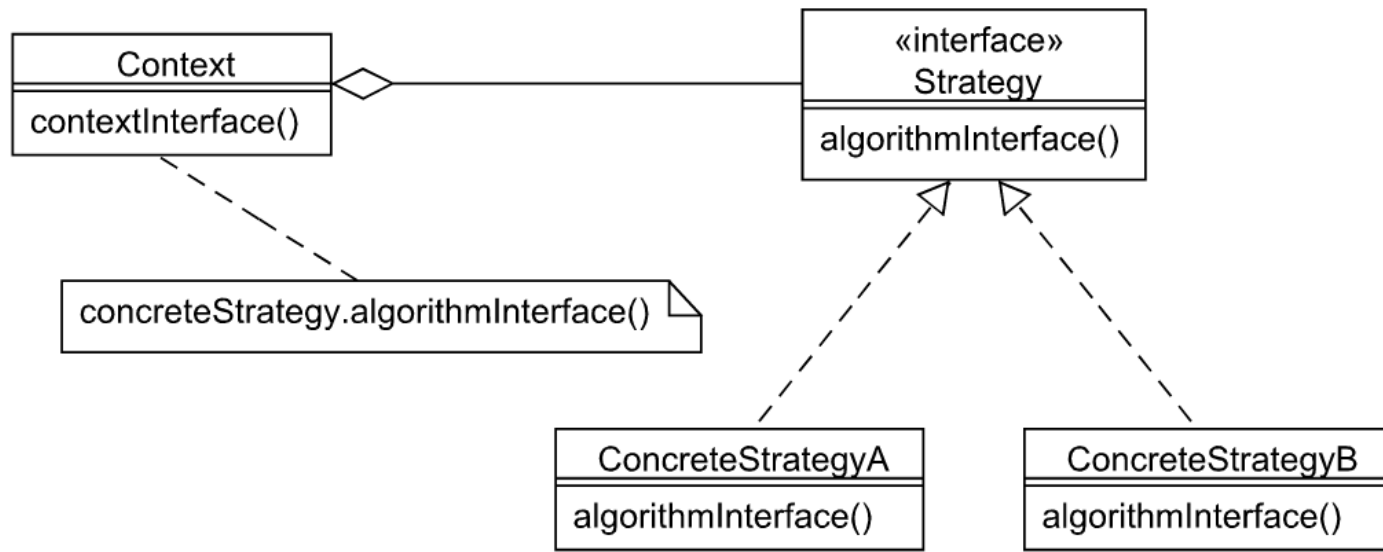
Events are handled by  
corresponding handlers  
in the context of the user  
interface elements

Events are handled by  
corresponding handlers  
in the context of the user  
interface elements

# Strategy

there are (and can be added) different  
strategies of solving a given problem,  
and the context of their application  
has to be ready for this





The interaction  
with GUI is based  
on capturing and  
handling events

A clash is an object. Its state changes.

GUI elements could be reacting differently to the change in the state of the clash, e.g., show the current overall energy.



A clash is an object. Its state changes.

GUI elements could be reacting differently to the change in the state of the clash, e.g., show the current overall energy.

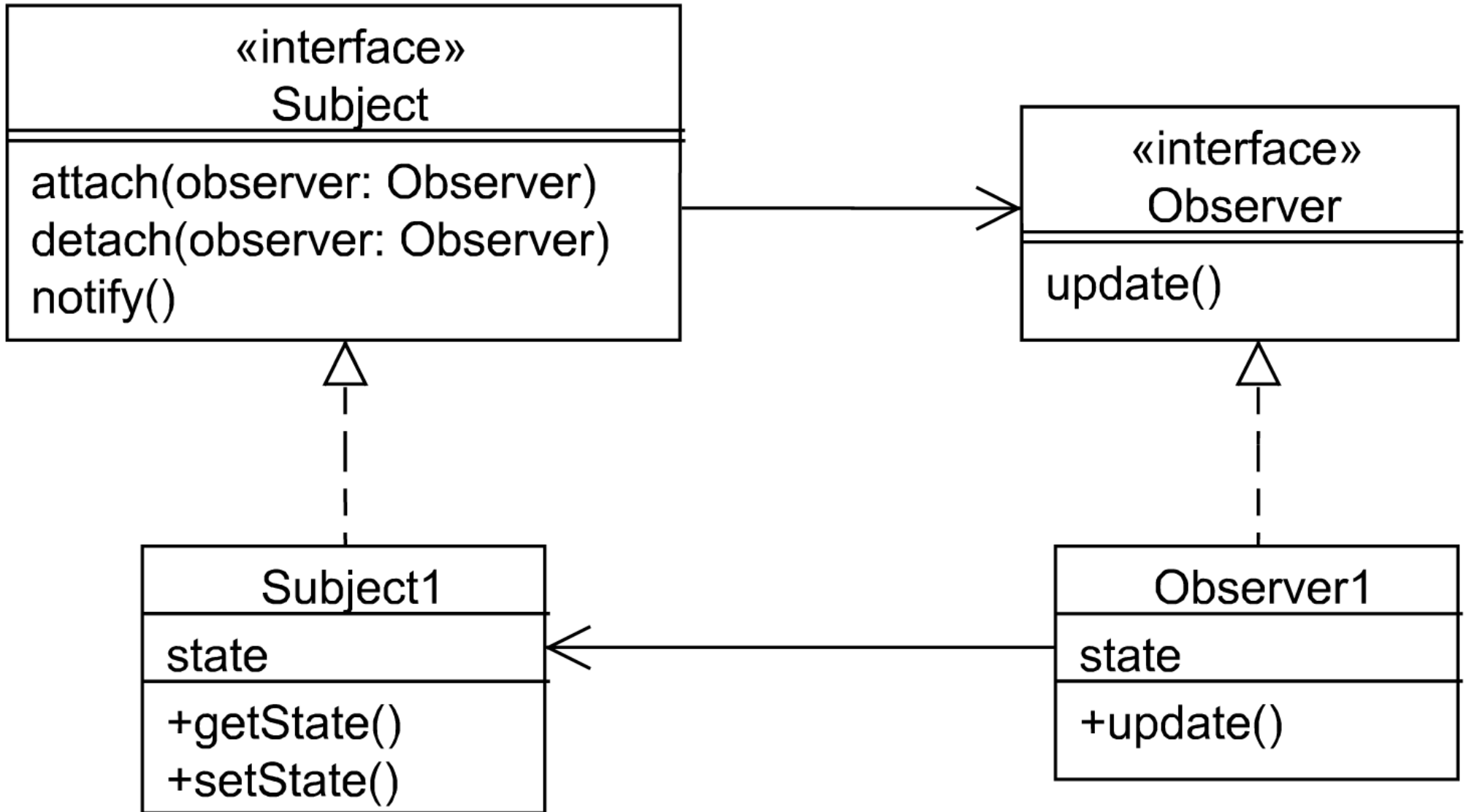
How to provide for adding such GUI elements without the need to change the clash object?

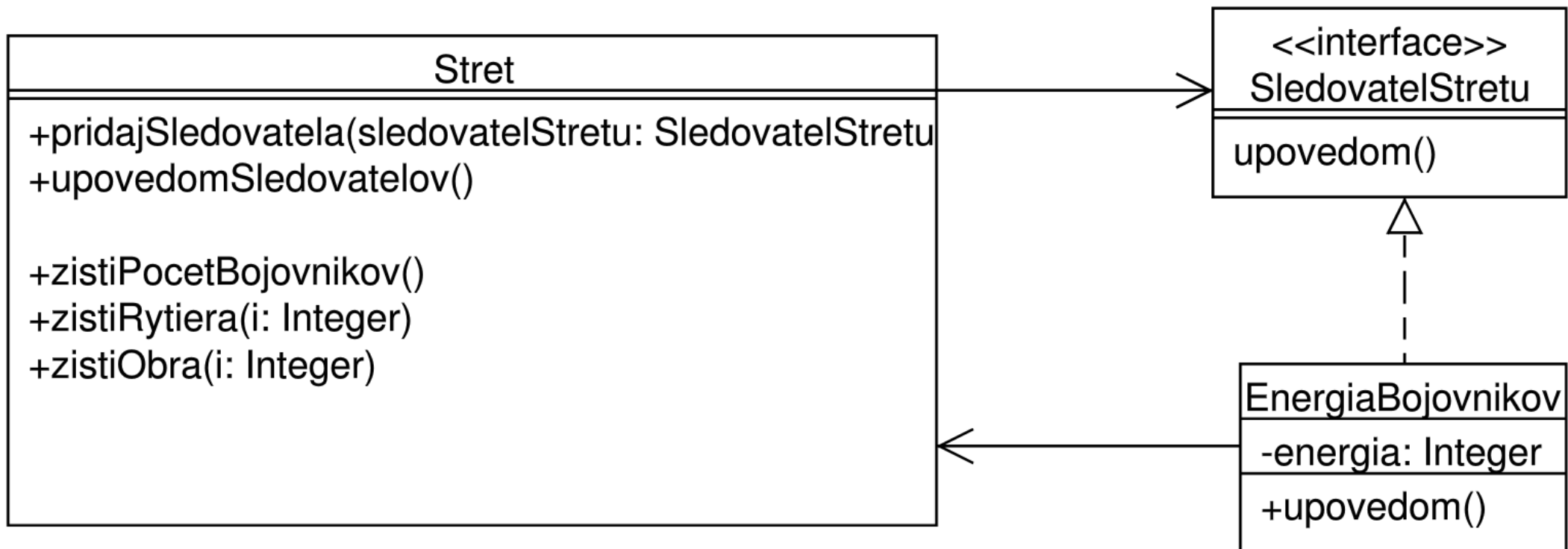
A user interface (or its parts) has to be notified of a change in the model, whereas it should be possible to add it without the need to change the model.

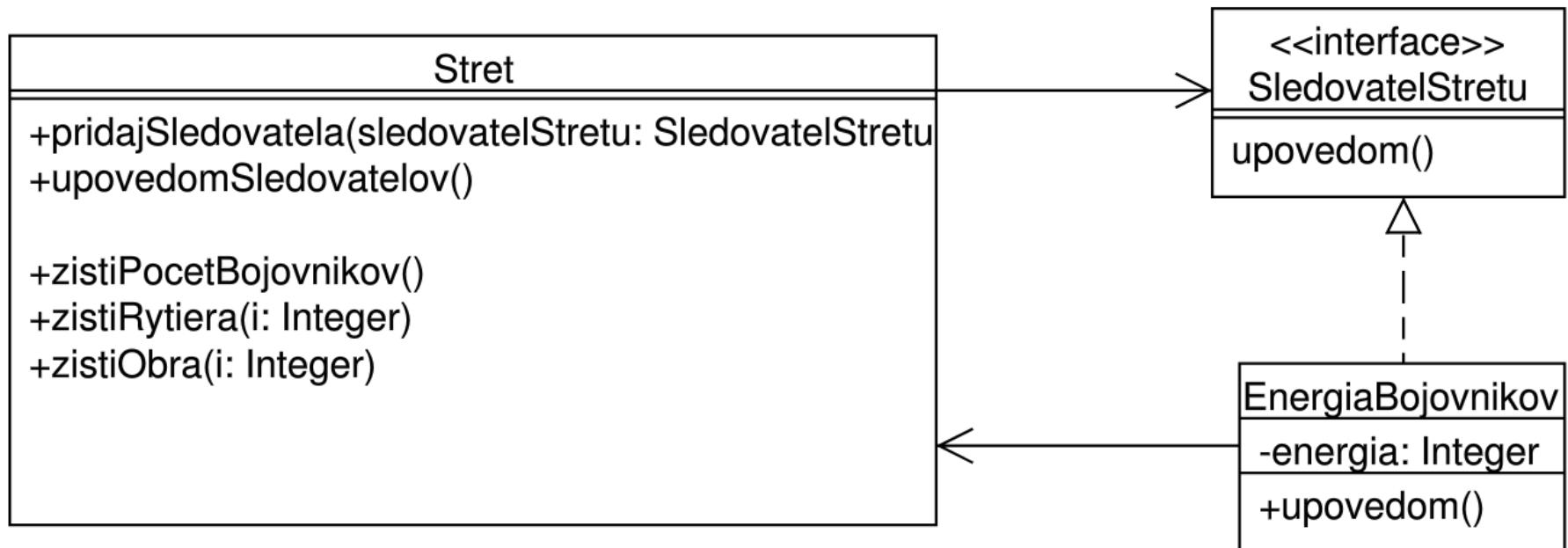
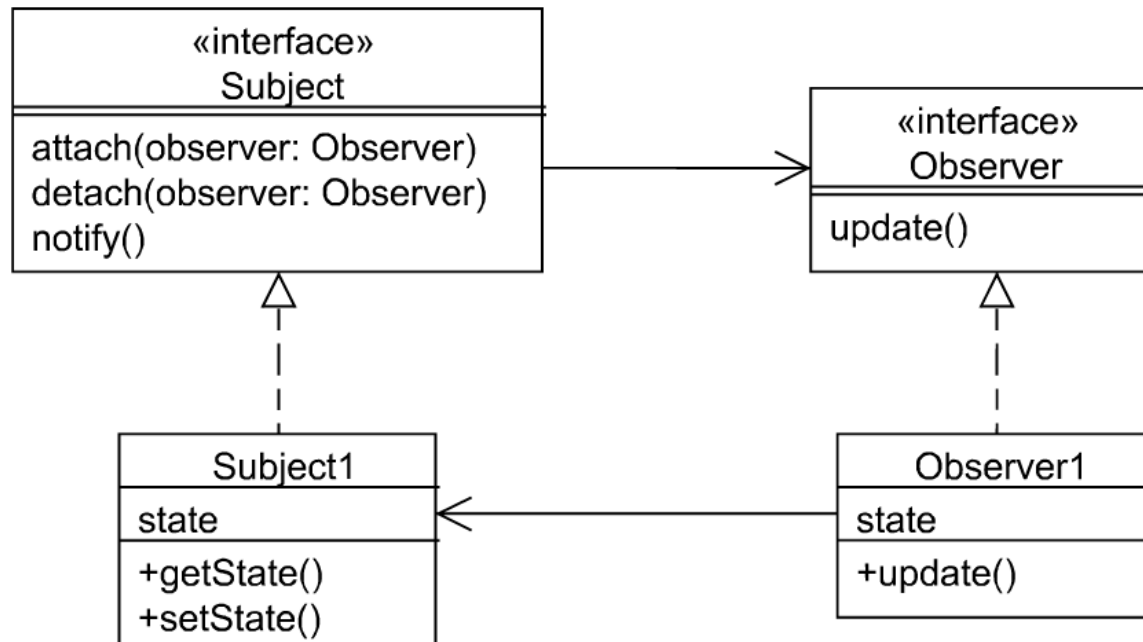
A user interface (or its parts) has to be notified of a change in the model, whereas it should be possible to add it without the need to change the model.

# Observer

observing objects have to be notified of a change in the state of the observed object, and it has to be possible to add them without a need to modify the observed object







Observer is  
appropriate for  
interconnecting the  
user interface and  
application logic

A user interface has to be decoupled from the application logic

The interaction with GUI is based on capturing and handling events

Observer is appropriate for interconnecting the user interface and application logic