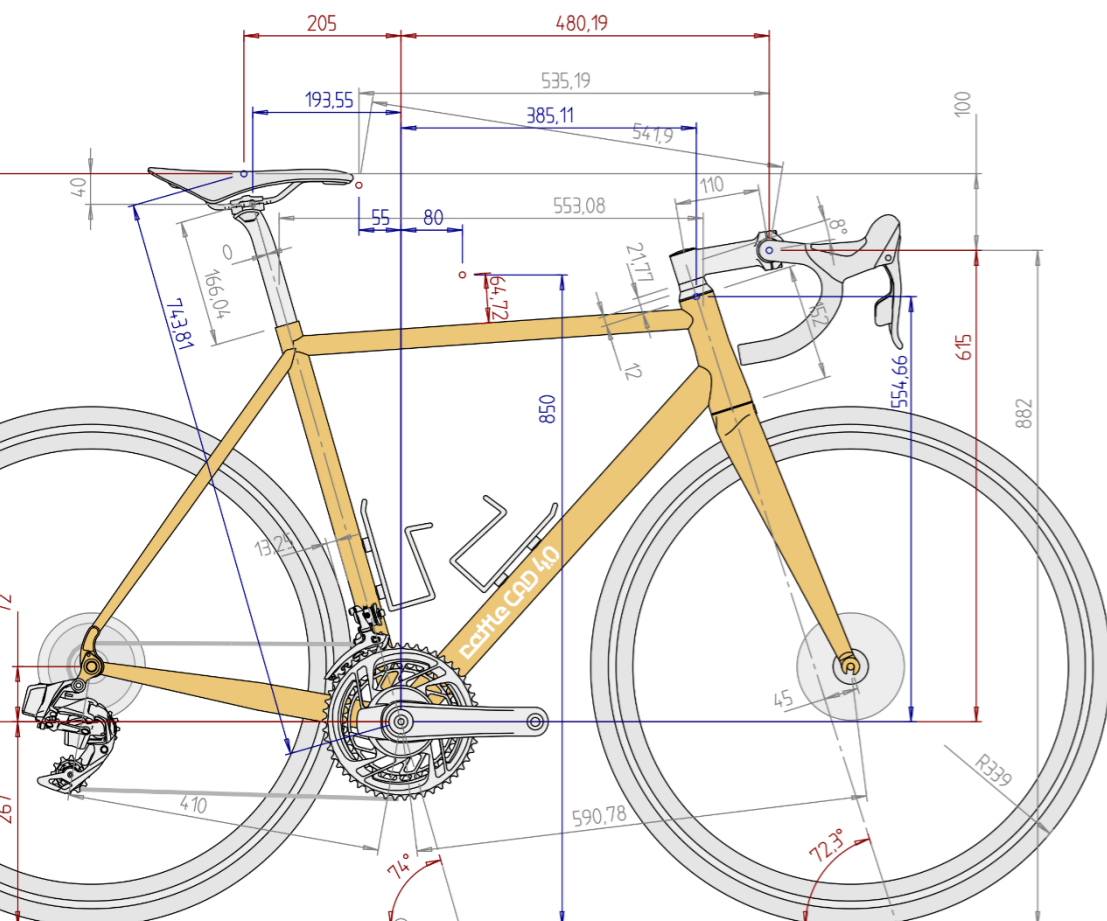




19th European TcI/Tk Conference

July 20th - 21st 2023, Vienna, Austria



rattle CAD

<https://rattleCAD.com/>
... design your custom Bike

Manfred ROSENBERGER

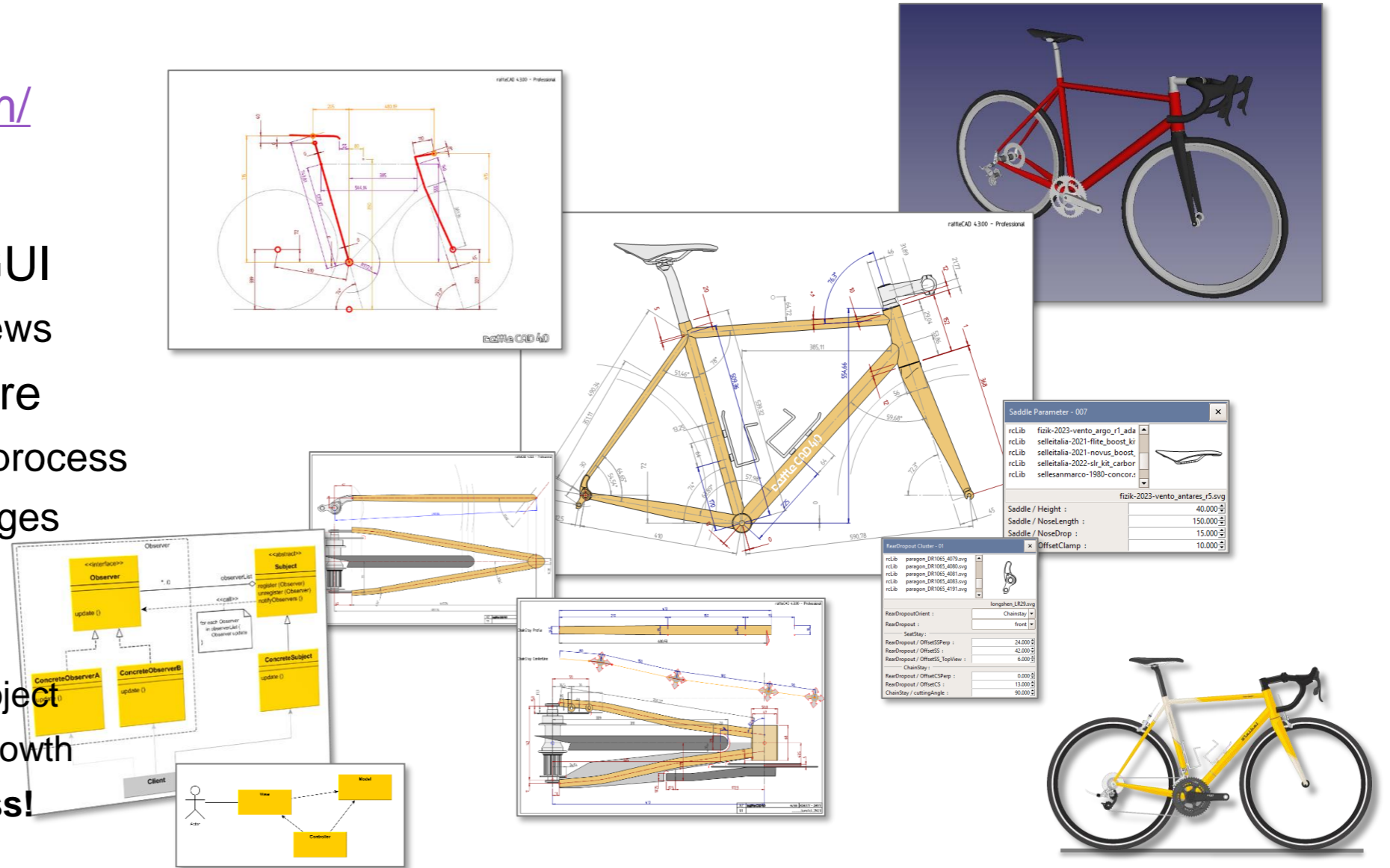
Implementation of the Observer Pattern with TclOO in the CAD software rattleCAD

rattleCAD 4.0

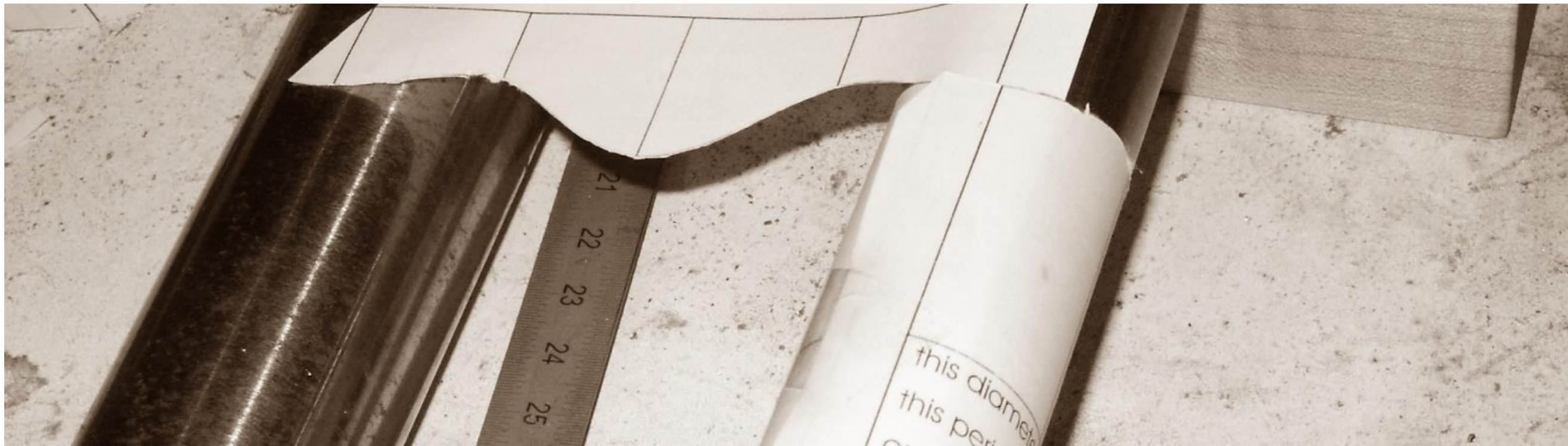
- <https://rattlecad.com/>

Context:

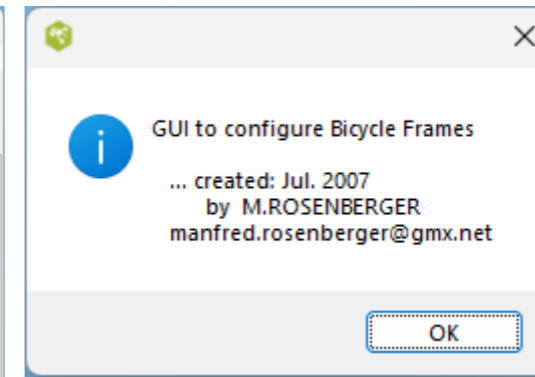
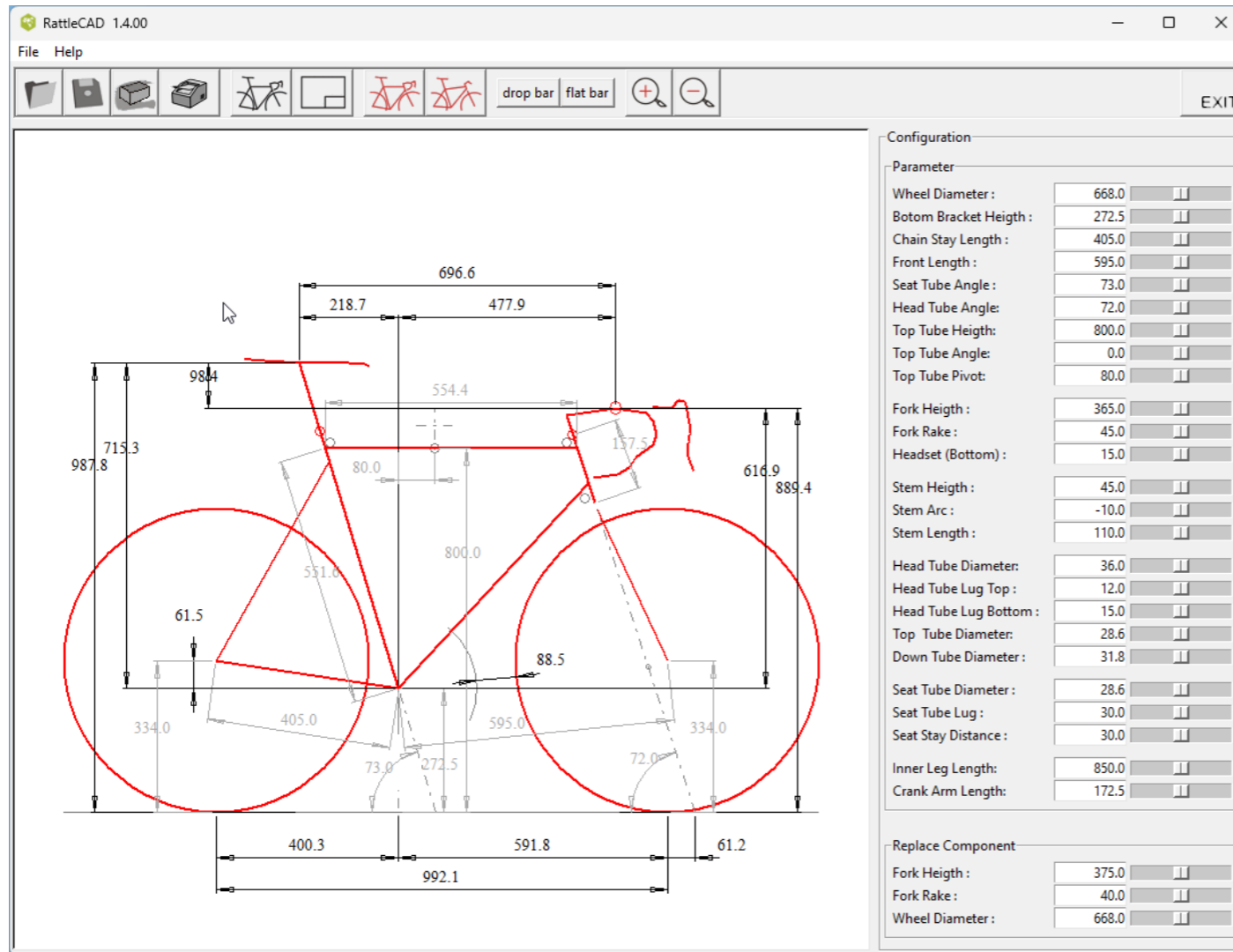
- User Interaction - GUI
 - manage various views
- Software Architecture
 - accelerate startup-process
 - independent packages
 - testability
 - improve quality!
 - start as small project
 - uncoordinated growth
 - cleanup the mess!



rattleCAD - Vintage Versions

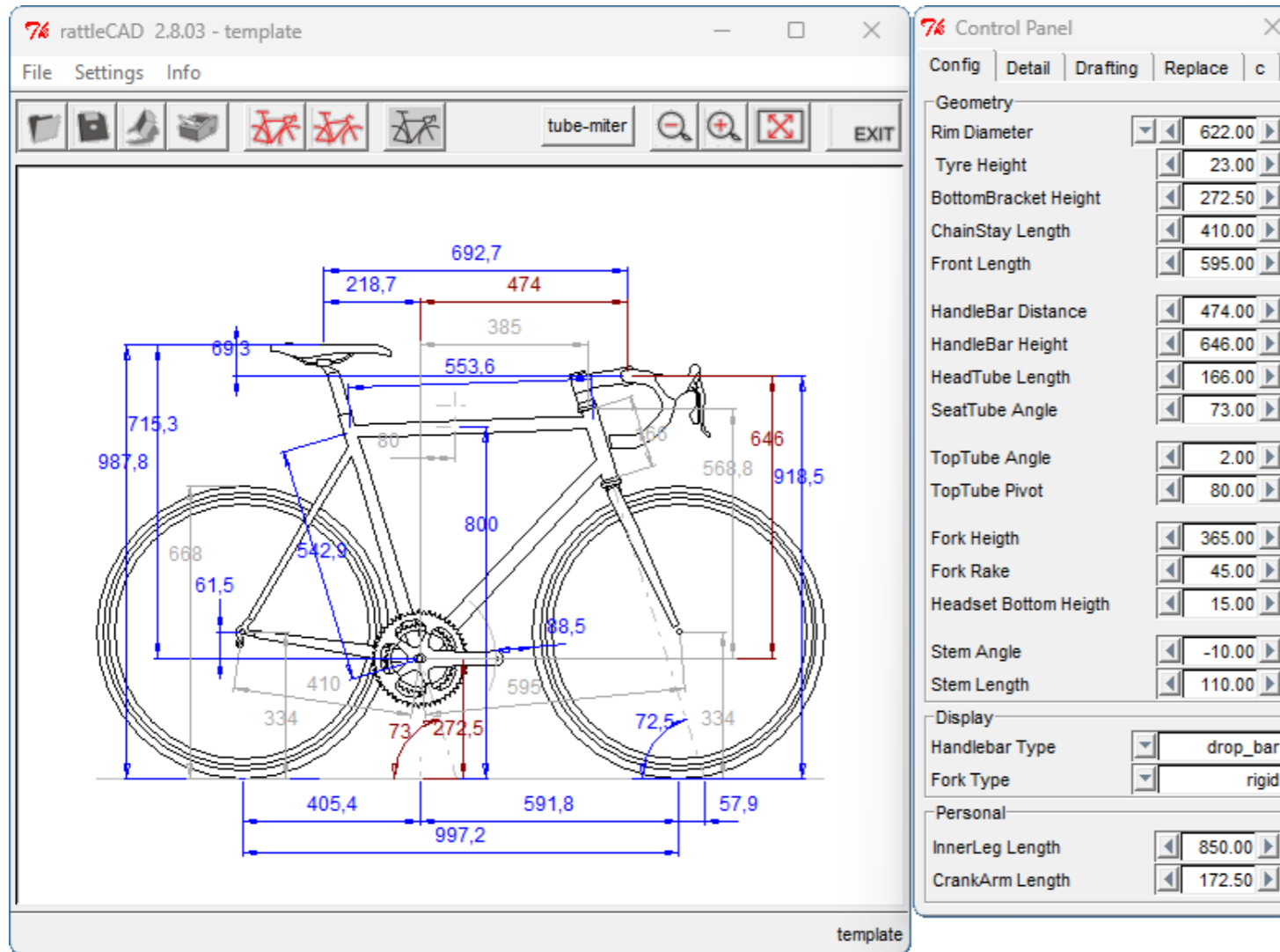


rattleCAD – History: 1.4.00 (2007)



Files: 46
Folder: 5
Size: 1,07 MB

rattleCAD – History: 2.8.03 (2009)



Files: 126
Folder: 12
Size: 1,16 MB

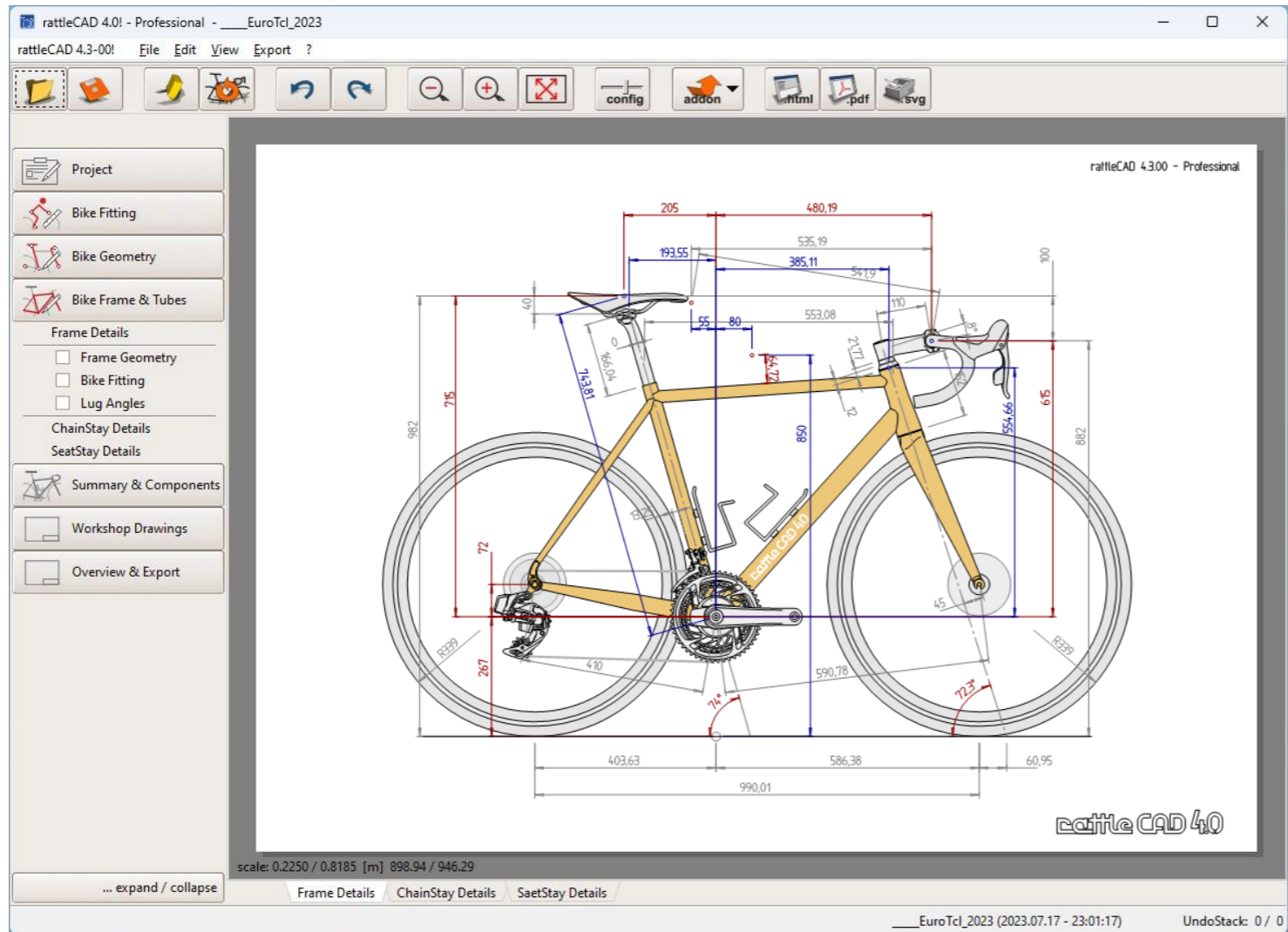
rattleCAD – History: 3.4.02 (2016)

The screenshot displays the rattleCAD 3.4.02.99 interface. The main window shows a technical drawing of a bicycle frame with various dimensions. The Configuration Panel on the right lists the following values:

Category	Parameter	Value
Base Concept - Primary Values	Geometry: HandleBar_Distance	481.00
	Geometry: HandleBar_Height	635.00
	Geometry: Saddle_Distance	212.69
	Geometry: Saddle_Height	710.00
	Geometry: Inseam_Length	850.00
	TopTube: PivotPosition	80.00
	SeatPost: Setback	25.00
Base Concept - Secondary Values	Geometry: Fork_Rake	45.00
	Geometry: Fork_Height	365.00
	Geometry: HeadTube_Angle	72.50
	Geometry: Stem_Angle	-6.00
	Geometry: Stem_Length	110.00
	Geometry: ChainStay_Length	410.00
	Geometry: BottomBracket_Depth	66.00
Base Concept - Alternative Values	Geometry: TopTube_LengthVirtual	555.889
	Geometry: FrontWheel_x	592.882
	Geometry: FrontWheel_xy	596.545
	Geometry: BottomBracket_Height	270.000
	Geometry: Saddle_BB	740.494
Wheels	Geometry: RearRim_Diameter	622.00
	Geometry: RearTyre_Height	25.00
	Geometry: FrontRim_Diameter	622.00
	Geometry: FrontTyre_Height	25.00

Files: 453
Folder: 70
Size: 15 MB

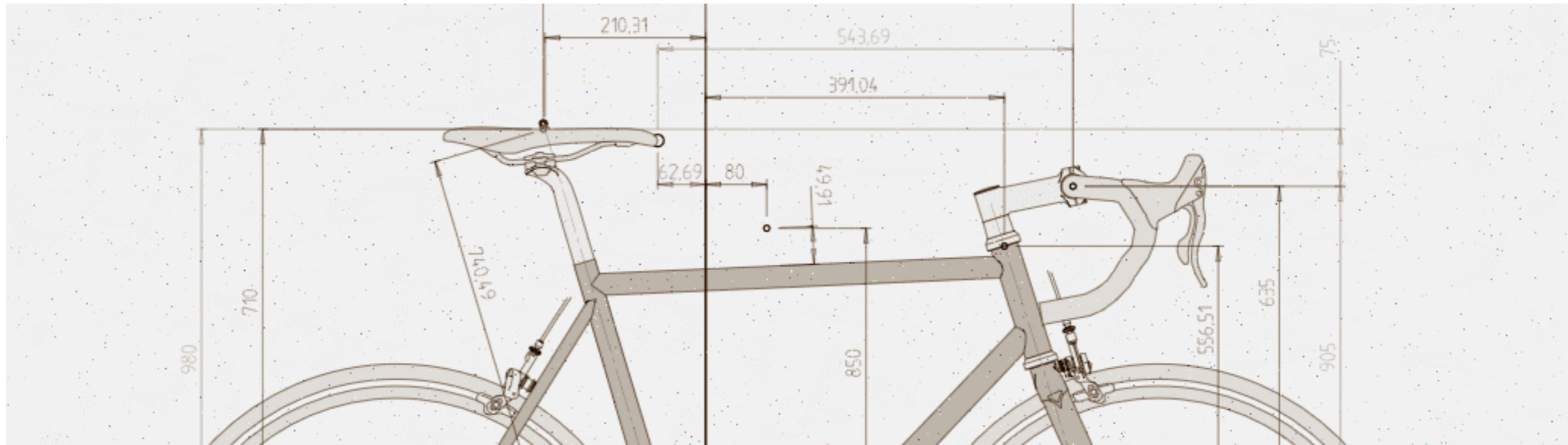
rattleCAD 4.0: 4.3.00 (2023, coming soon)



Files: 1614
Folder: 189
Size: 14,6 MB

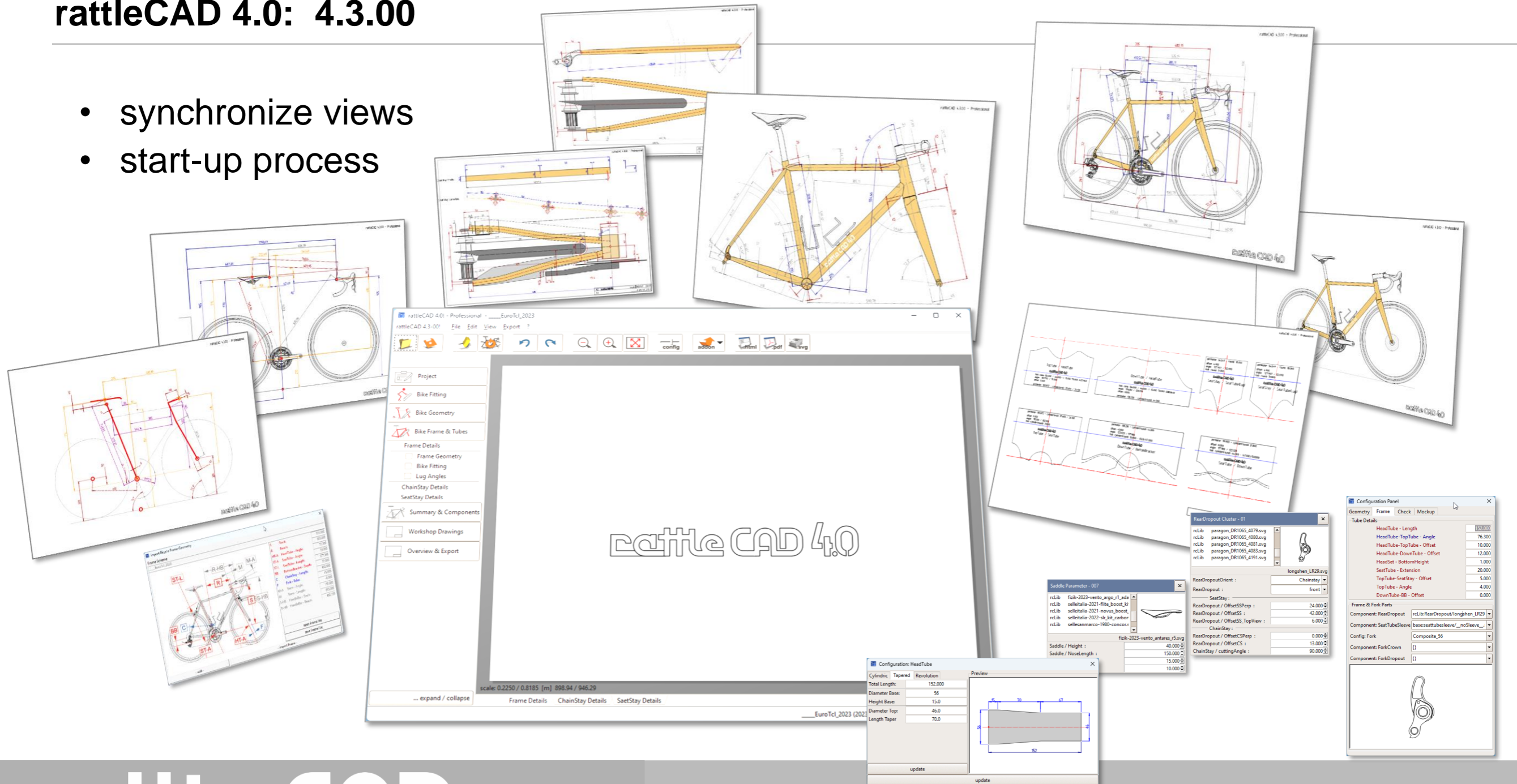
component files in an external library: 7.5 MB

Challenges of current Improvements



rattleCAD 4.0: 4.3.00

- synchronize views
- start-up process



rattleCAD 4.0

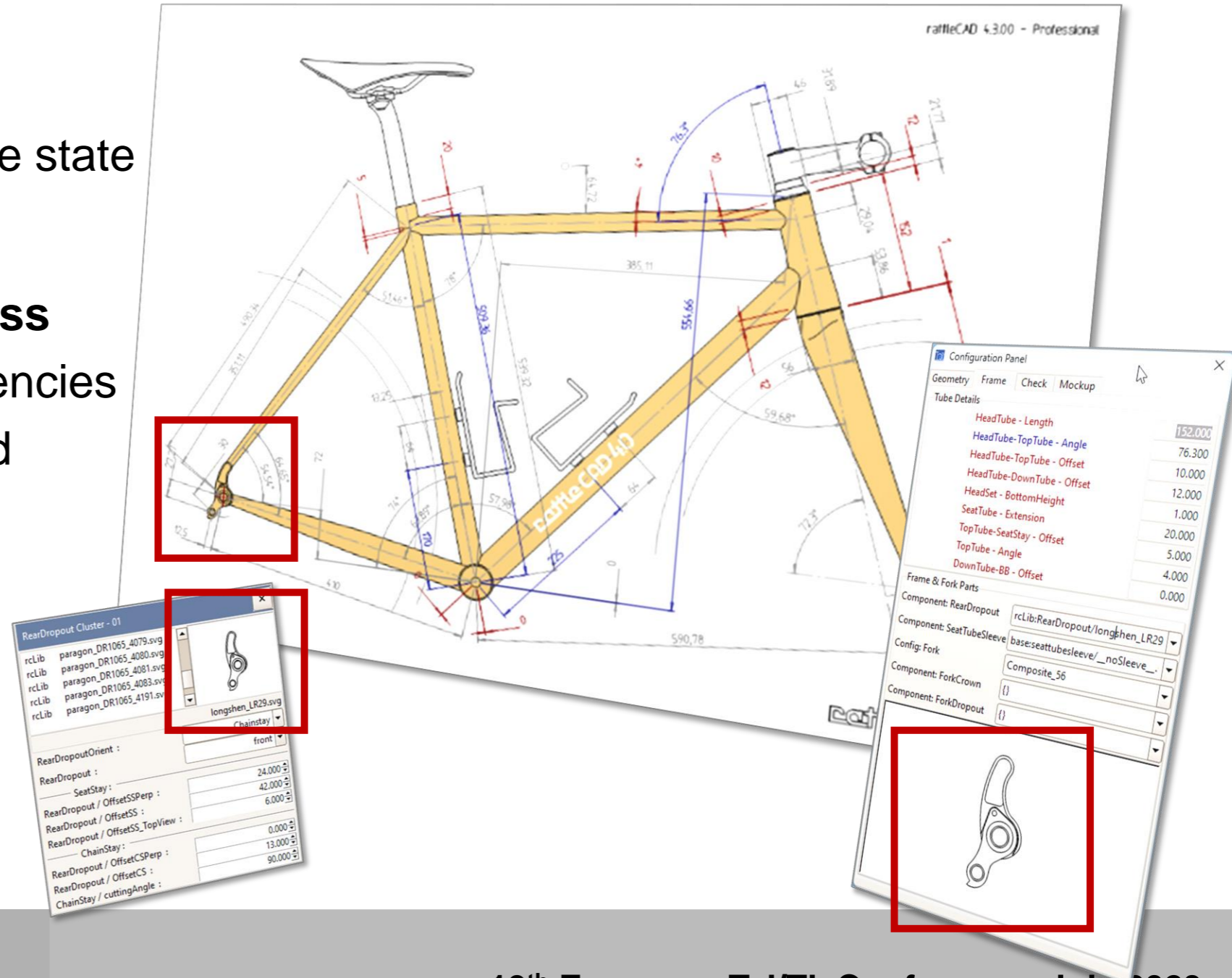
rattleCAD 4.0: Challenge

Synchronize Views

- every view shows the same state

Accelerate the Start-up Process

- complex package dependencies
- fully loaded model required
- multiple view updates



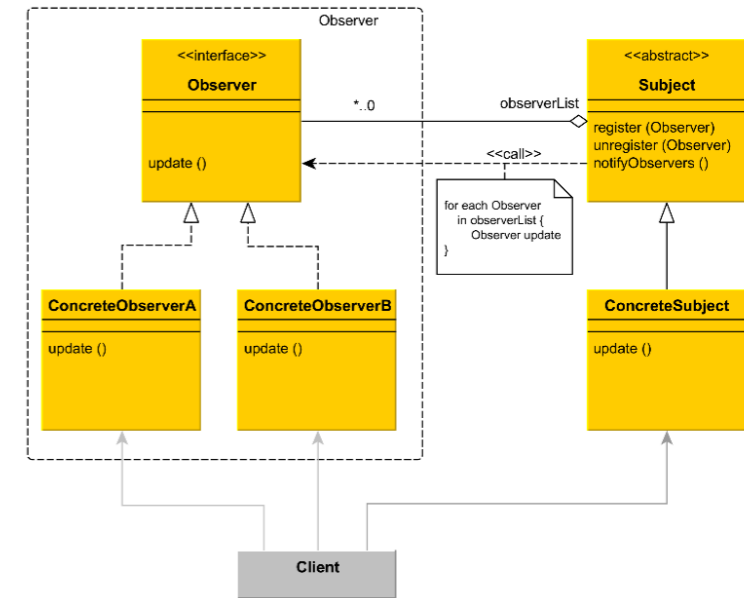
Approach



Approach

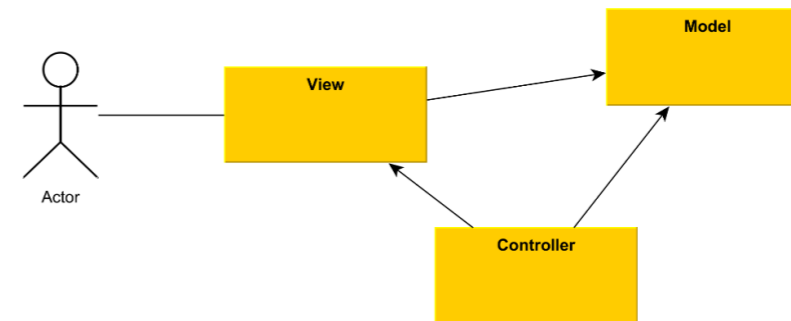
Approach

- divide and conquer
 - independent packages
 - separate data, GUI and logic
- Concepts & Strategies
 - **MVC Pattern**
 - **Observer Pattern**

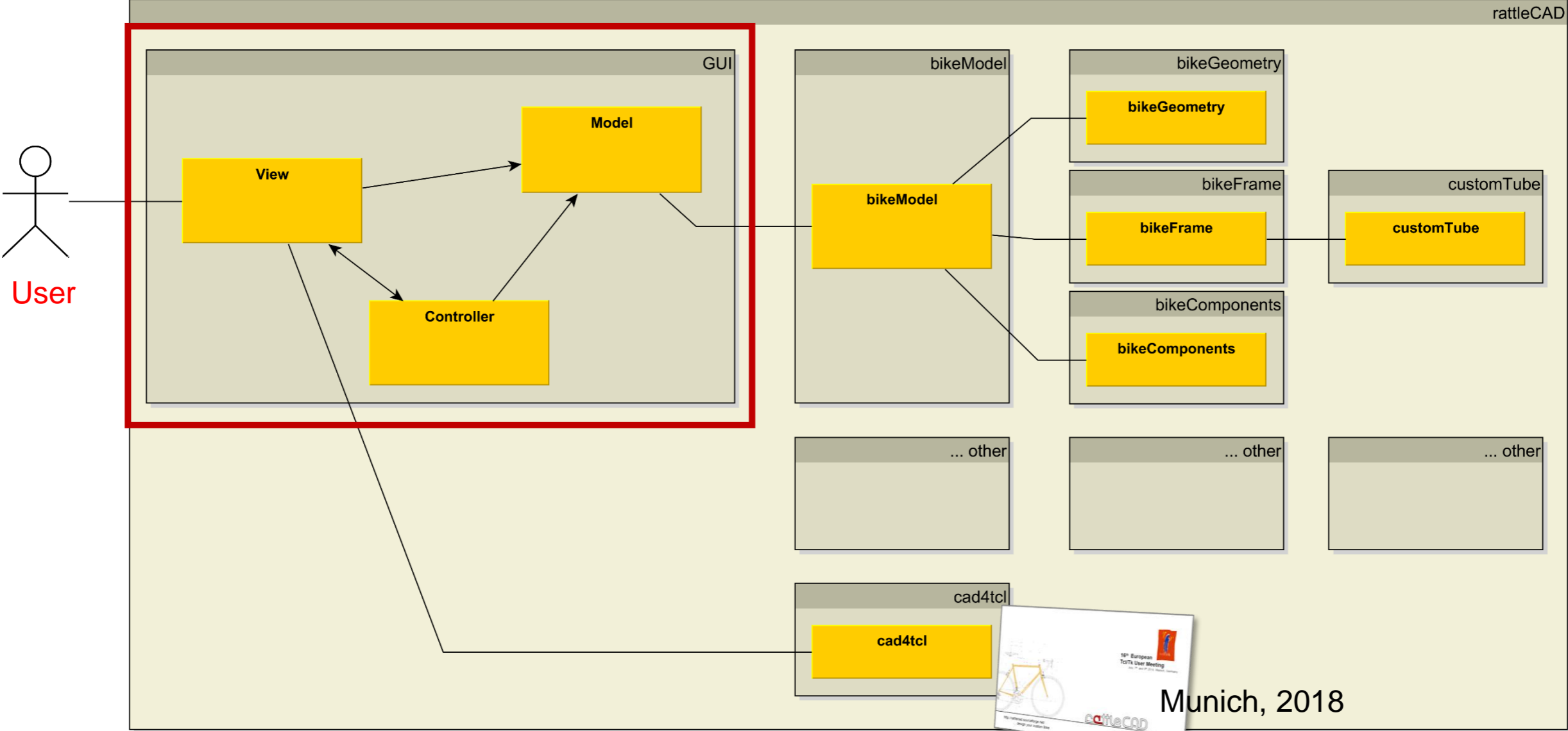


Refactor Code Base (neverending story)

- improve Software architecture
- improve testability
 - test model independent from the GUI
 - initialize the GUI without loaded model

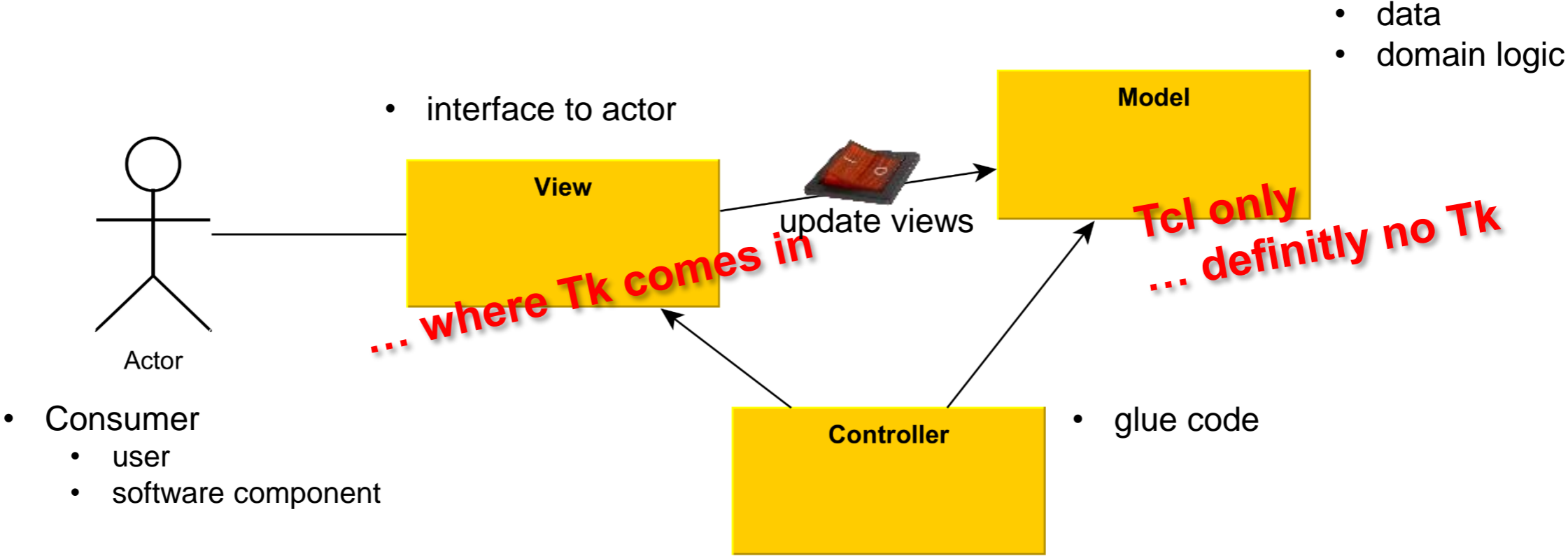


rattleCAD 4.0 - Packages



Munich, 2018

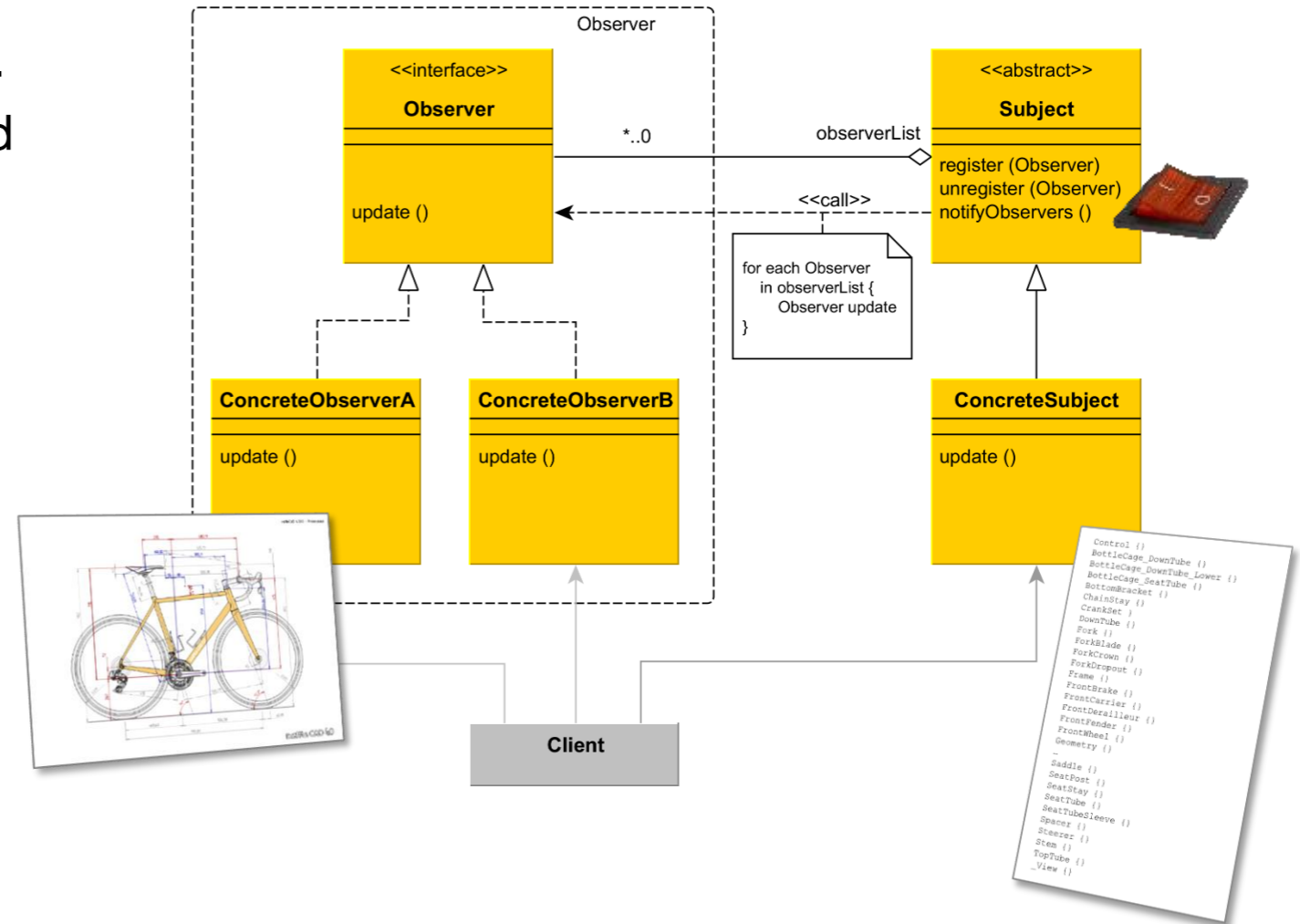
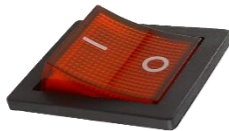
MVC - Pattern



The Observer Pattern (Gang of Four)

Definition

- 1-to-n dependency between objects.
- A change in the state of an observed object causes all dependent objects to be notified and automatically updated. ([GoF], page 287)
- Objects (observer, observing object) can be registered by another object (subject, observed object) and henceforth be informed by the latter as soon as it changes.
- This registration can be cancelled.



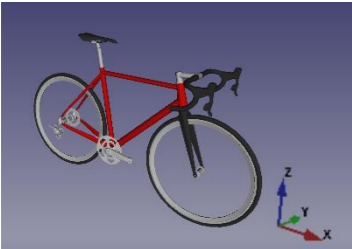
<https://www.philippauer.de/study/se/design-pattern/observer.php#gof>

[GoF] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: "Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software". Addison-Wesley. 1. Auflage 1996.

Manage View Updates with the Observer-Pattern



rattleCAD 4.0 - Views



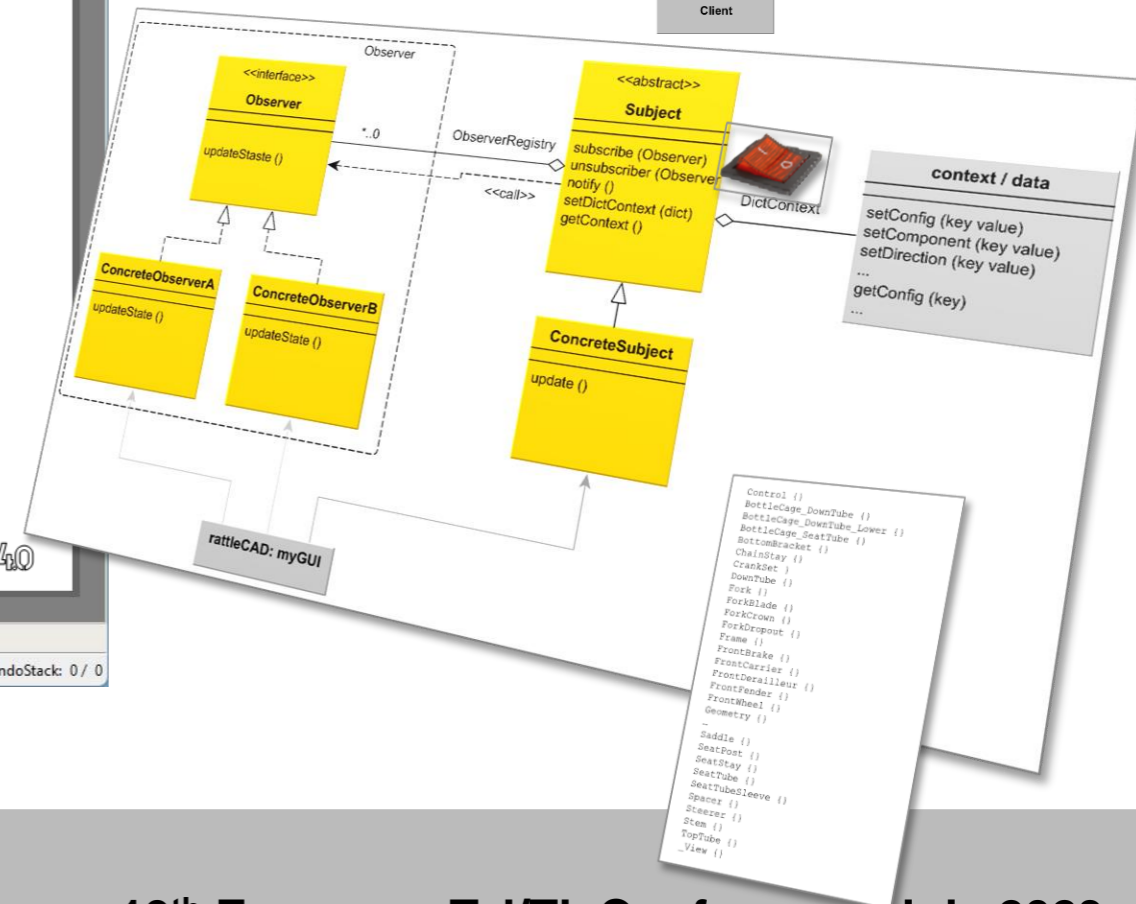
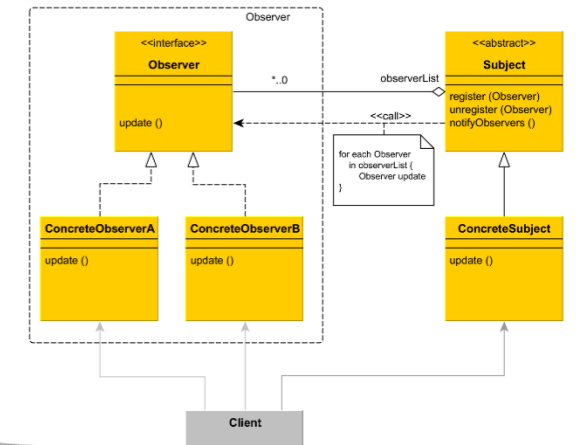
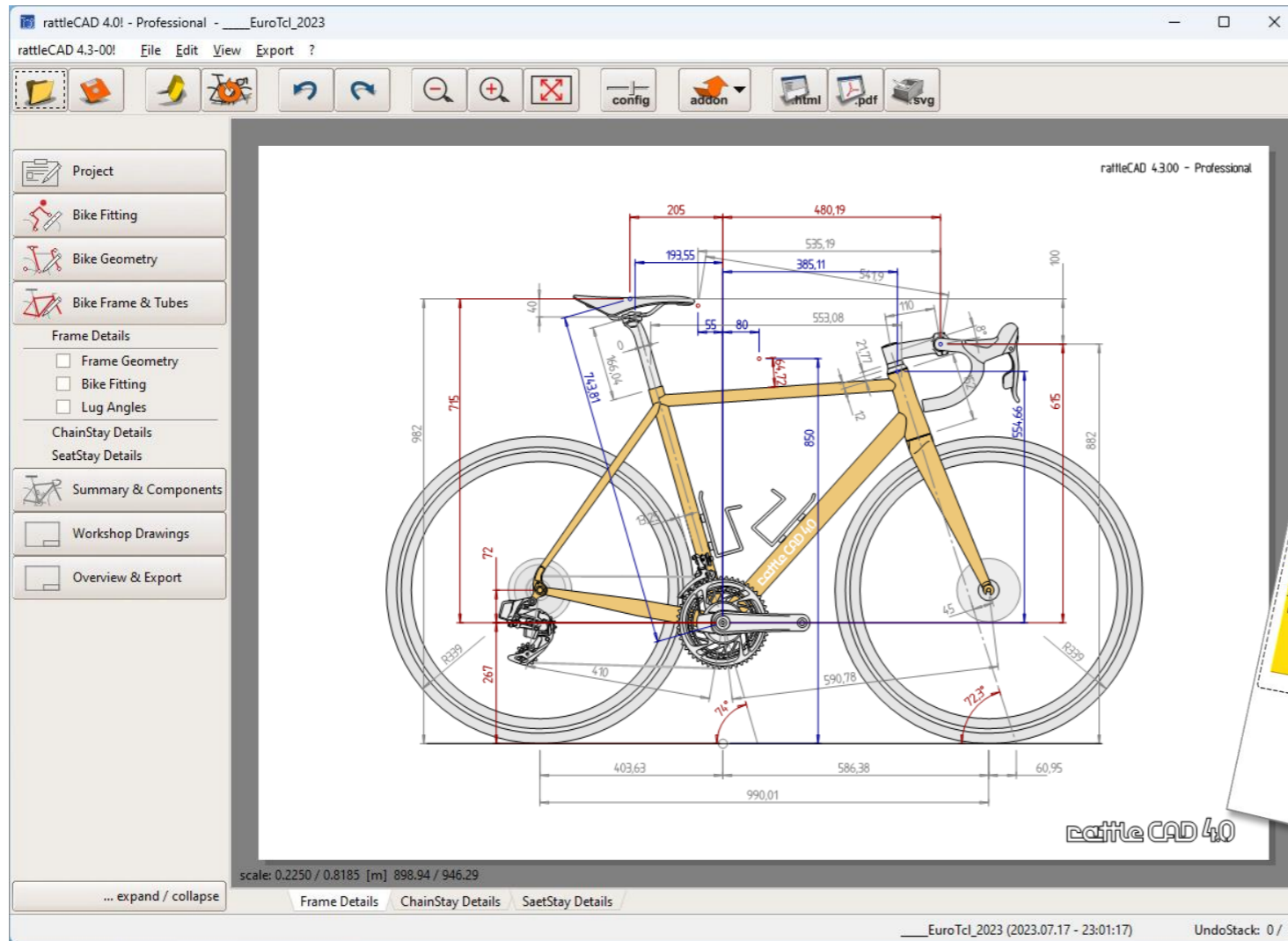
The collage displays multiple views of a bicycle frame in rattleCAD 4.0. It includes:

- 2D technical drawings of the frame tubes and components.
- 3D perspective and isometric views of the frame.
- Configuration panels for various parts like the head tube, rear dropout, and saddle.
- A central window showing the software interface with a list of views on the left and a central workspace.

Context:

- XZ
- XY
- Edit
- TubeMiter

Observer Pattern in the context of the rattleCAD project



```

Control ()
BottleCage_DownTube ()
BottleCage_DownTube_Lower ()
BottleCage_SeatTube ()
BottomBracket ()
ChainStay ()
CrankSet ()
DownTube ()
Fork ()
ForkBlade ()
ForkCrown ()
ForkDropout ()
Frame ()
FrontBrake ()
FrontCarrier ()
FrontDerailleur ()
FrontFender ()
FrontWheel ()
Geometry ()
Saddle ()
SeatPost ()
SeatStay ()
SeatTube ()
SeatTube ()
SeatTubeSleeve ()
Spacer ()
Stem ()
Stem ()
TopTube ()
_View ()

```

Observer Pattern in the context of the rattleCAD project - Subject

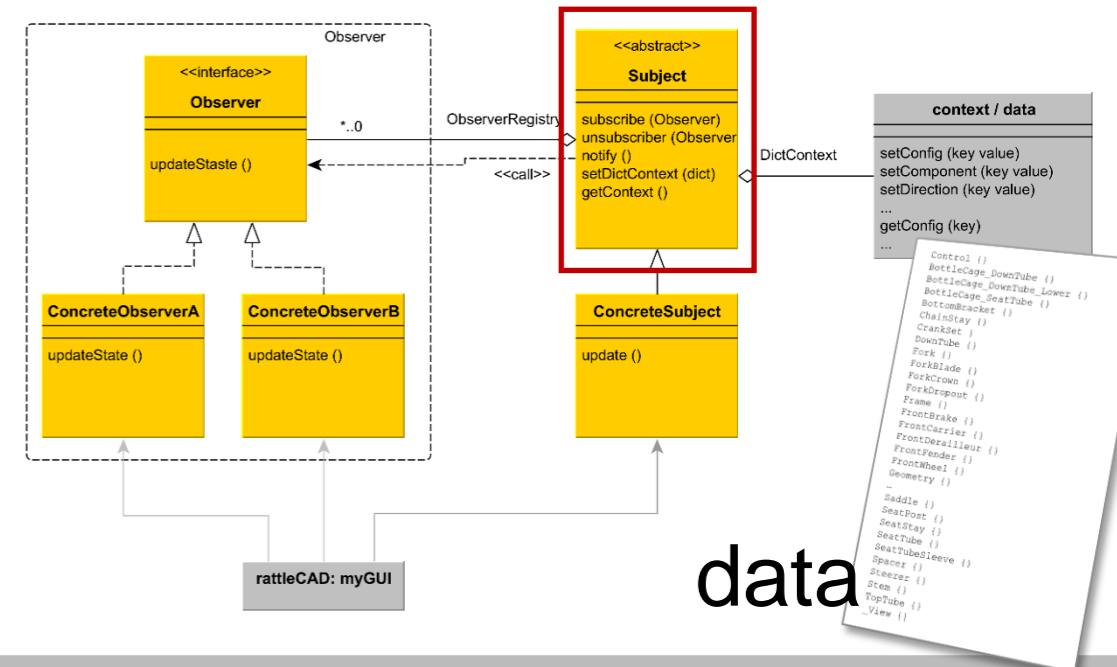
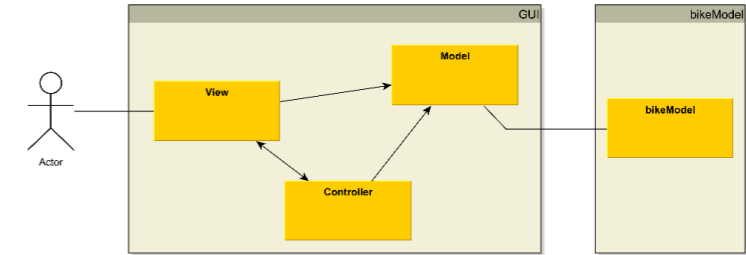
Subject:

```

oo::class create Subject {

    variable ObserverRegistry
    variable DictContext

    method subscribe {observer} {
        lappend ObserverRegistry $observer
    }
    #
    method unsubscribe {observer} {
        set idx [lsearch $ObserverRegistry $observer]
        set ObserverRegistry [lreplace $ObserverRegistry $idx $idx]
    }
    #
    method notify {{state {}}} {
        set StateNotificaton $state
        foreach subscriber $ObserverRegistry {
            $subscriber updateState [self]
        }
    }
    #
    method setDictContext {dictContext} {
        set DictContext $dictContext
    }
    #
    method getContext {{key {}}} {
        if [dict exists $DictContext $key] {
            return [dict get $DictContext $key]
        }
    }
}
    
```

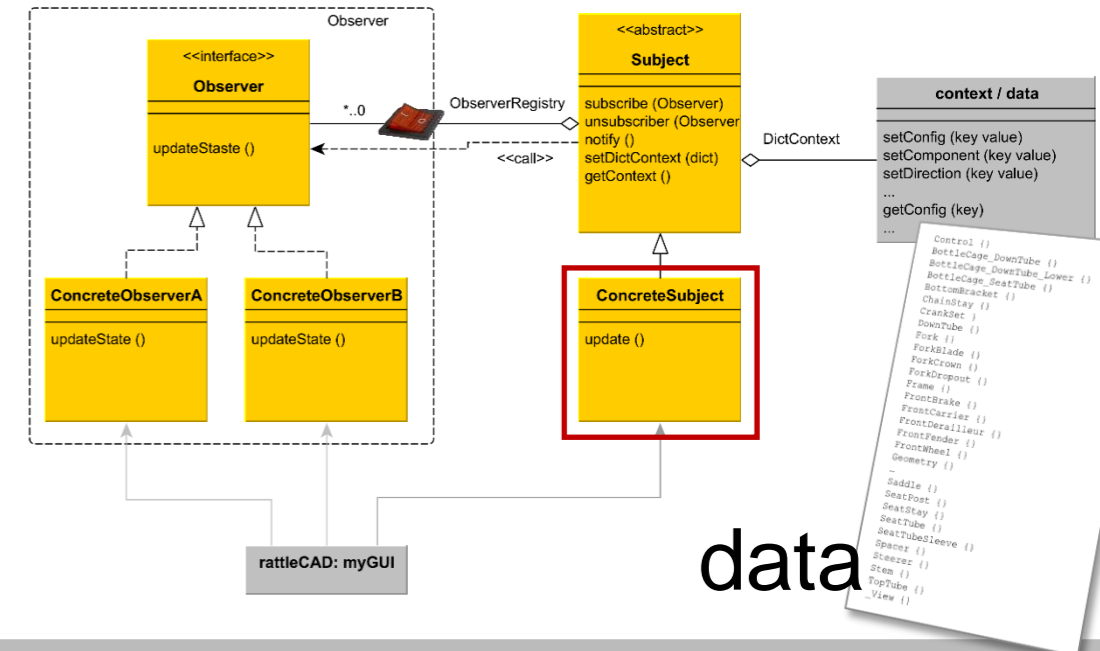
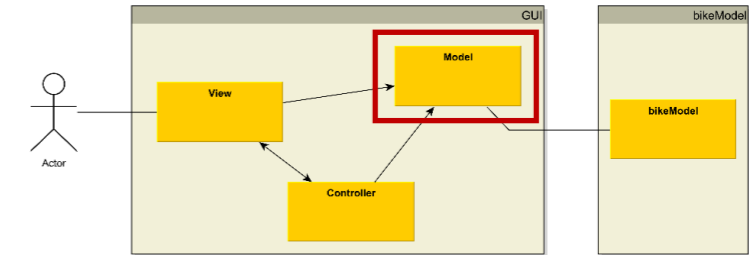


Observer Pattern in the context of the rattleCAD project - Concrete Subject

Concrete Subject:

```

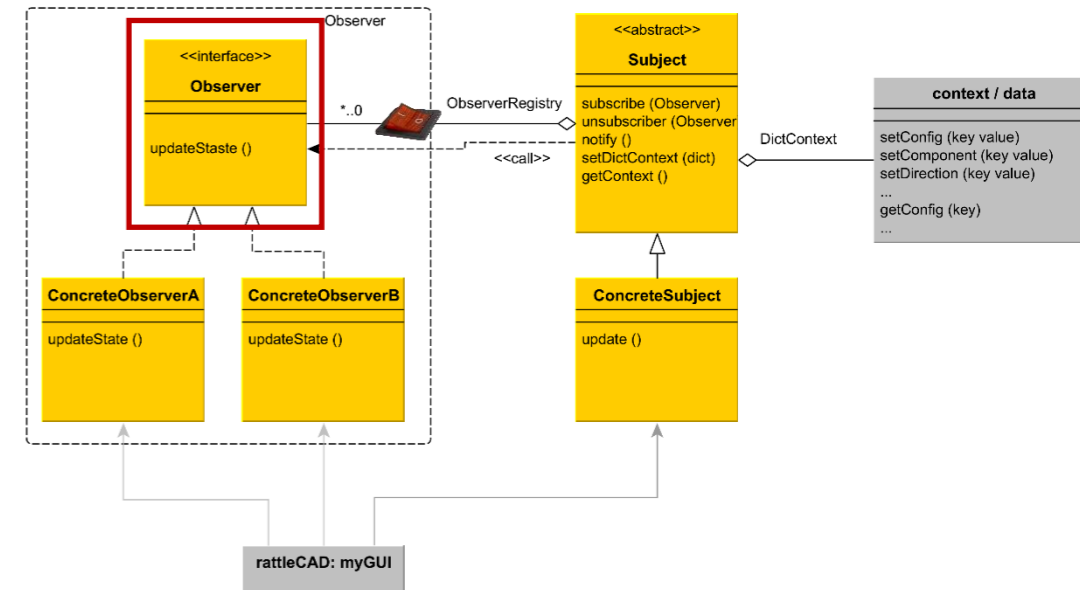
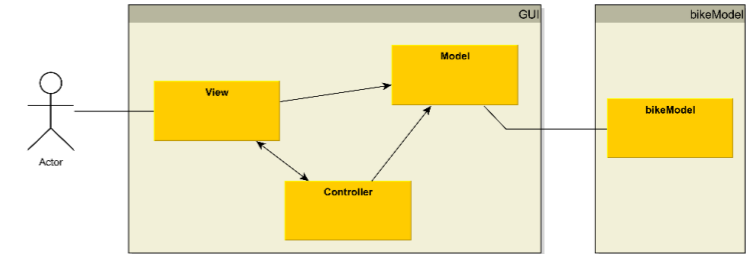
proc myGUI::modelBike::init {} {
    #
    variable Object_BikeModel [IF_BikeModel new]
    variable State_BikeModel [State_BikeModel new]
    #
    variable Subject_ProjectInfo [Subject new]
    variable Subject_BikeModel [Subject new]
    variable Subject_BikeModel_Edit [Subject new]
    #
    myGUI::modelBike::model_Edit::setListBoxDict \
        [$Object_BikeModel get_ListBoxValues]
    #
    $Subject_BikeModel setDictContext \
        {
            xy ::myGUI::modelBike::model_XY
            xz ::myGUI::modelBike::model_XZ
            tubeMiter ::myGUI::modelBike::model_TubeMiter
        }
    #
    $Subject_BikeModel_Edit setDictContext \
        {
            edit ::myGUI::modelBike::model_Edit
        }
    #
}
    
```



Observer Pattern in the context of the rattleCAD project - Observer

Observer:

```
oo::class create Observer {  
  
  method updateState {subject} {  
    puts ""  
    puts "  -> [info object class [self]] -> updateState:"  
    puts "  -> subject: $subject"  
    set data [$subject getData]  
    puts "  -> data: $data"  
    puts ""  
    puts "... to be overwritten by custom observer class"  
    puts ""  
  }  
}
```

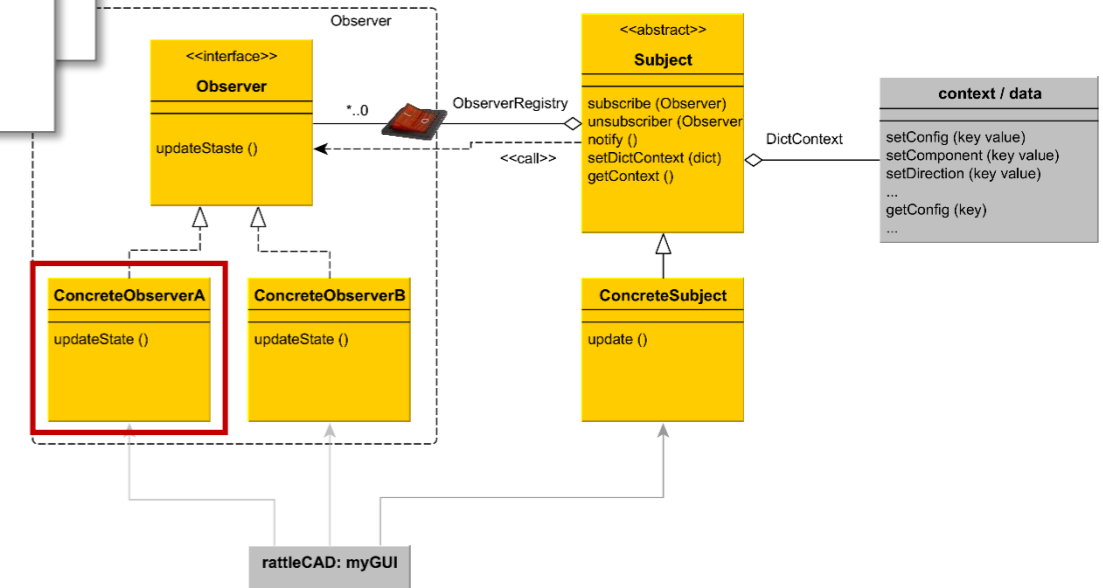
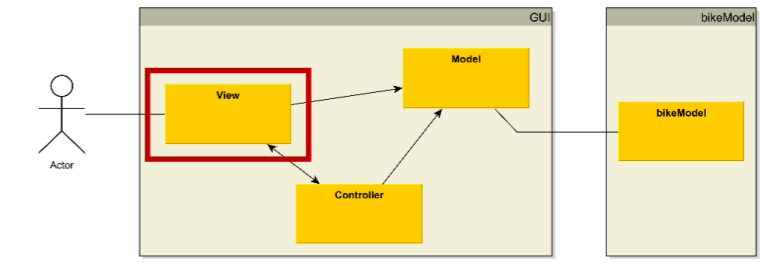
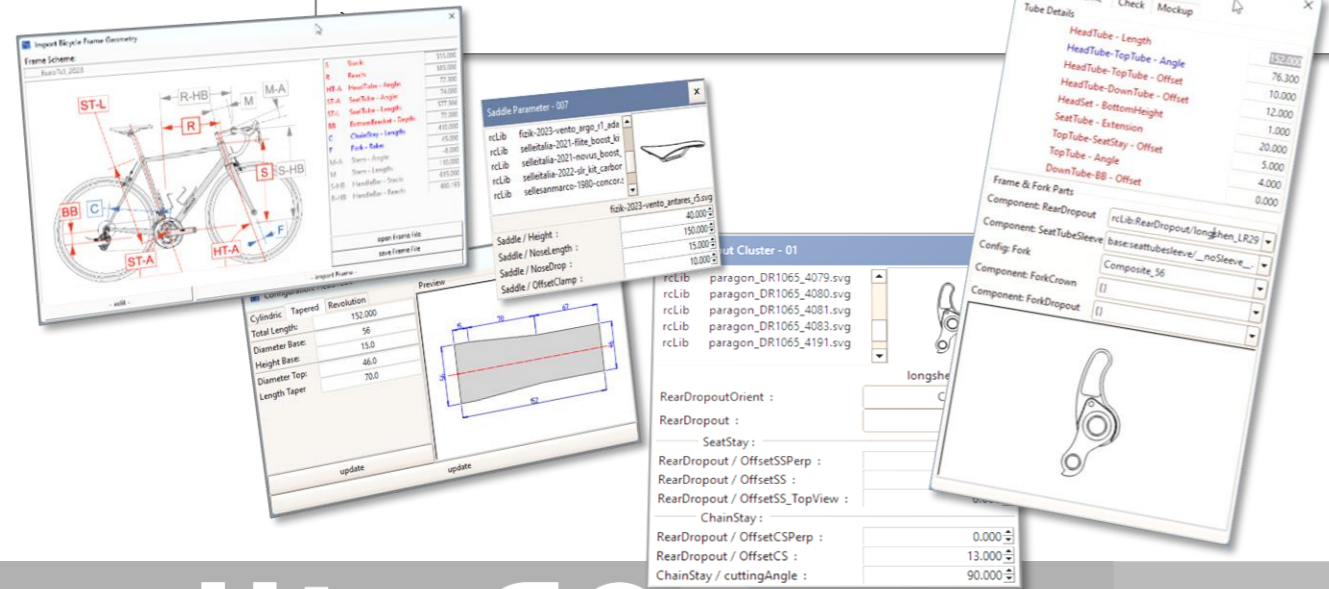


Observer Pattern in the context of the rattleCAD project - Concrete Observer

Concrete Observer:

```
oo::class create myGUI::viewDrawing::ObserverDrawingView {
#
oo::class create myGUI::viewEdit::ObserverEditView {
#
superclass Observer

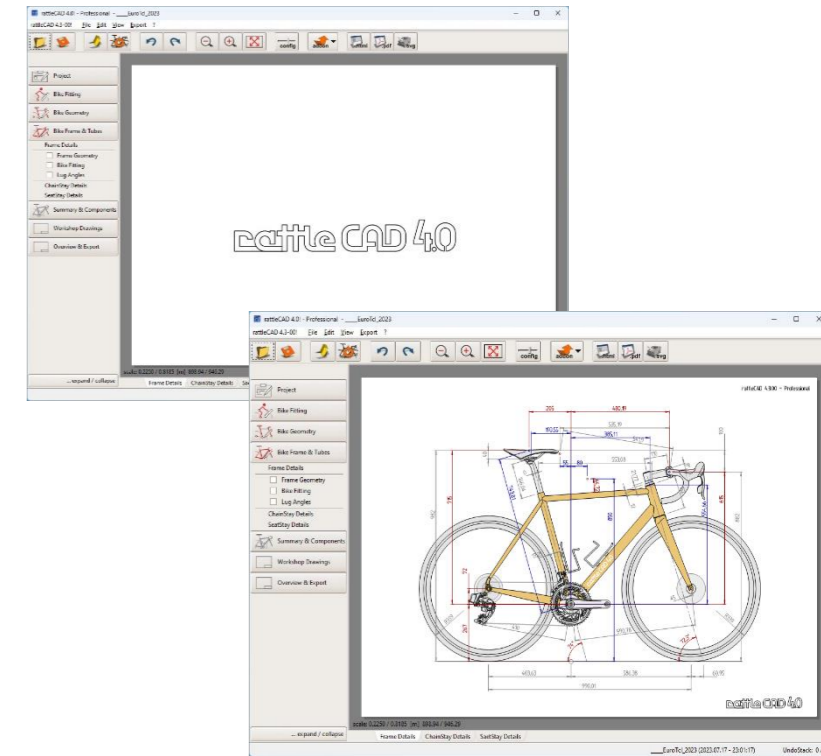
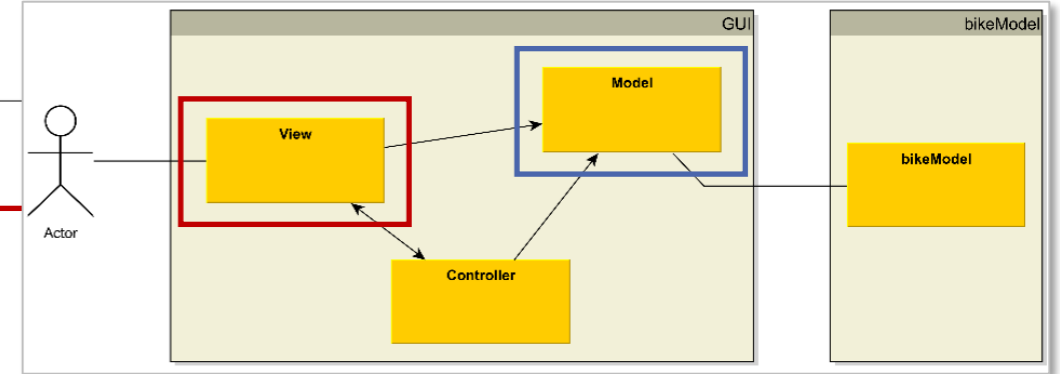
method updateState {subject args} {
#
set contextSubject_Edit [$subject getContext edit]
#
set myGUI::viewEdit::SubjectModel_Edit $contextSubject_Edit
#
myGUI::viewEdit::updateCurrentValues
#
}
```



rattleCAD Startup Routine

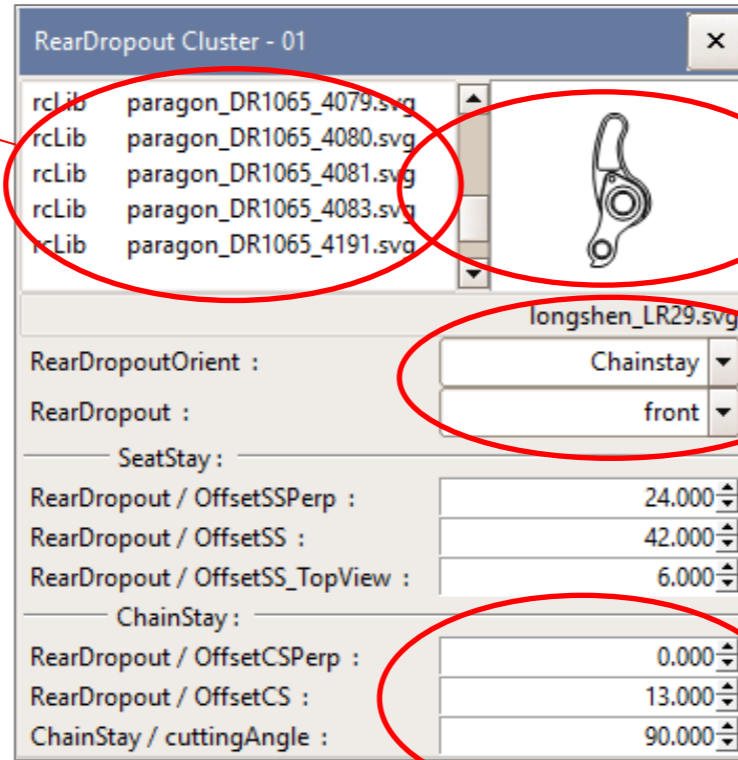
Switch on the Update-Process late

```
myGUI::main {rootDir baseDir {startupProject {}}} {  
  # -- GUI  
  #  
  myGUI::init $rootDir $baseDir  
  myGUI::create_intro .intro  
  myGUI::modelSession::update_windowTitle  
  myGUI::init_GUI  
  myGUI::update_GUI_Bindings  
  myGUI::gui::set_AccordionMainTab frametubes  
  myGUI::gui::show_CanvasLogo  
  ...  
  #  
  # -- Model  
  #  
  myGUI::modelBike::init  
  #  
  myGUI::modelBike::update_UserCompDirectories  
  myGUI::modelBike::update_UserTemplateDirecty  
  myGUI::modelBike::add_ComponentDir user ...  
  myGUI::modelBike::add_ComponentLib rcLib ...  
  ...  
  #  
  # -- Model - Binding  
  #  
  set observerDrawingView [myGUI::viewDrawing::ObserverDrawingView new  
  set observerEditView [myGUI::viewEdit::ObserverEditView new]  
  #  
  $::myGUI::modelBike::Subject_BikeModel subscribe $observerDrawingView  
  $::myGUI::modelBike::Subject_BikeModel_Edit subscribe $observerEditView  
  ...  
  #  
}
```



View-Widgets in myGUI::viewEdit

treeview

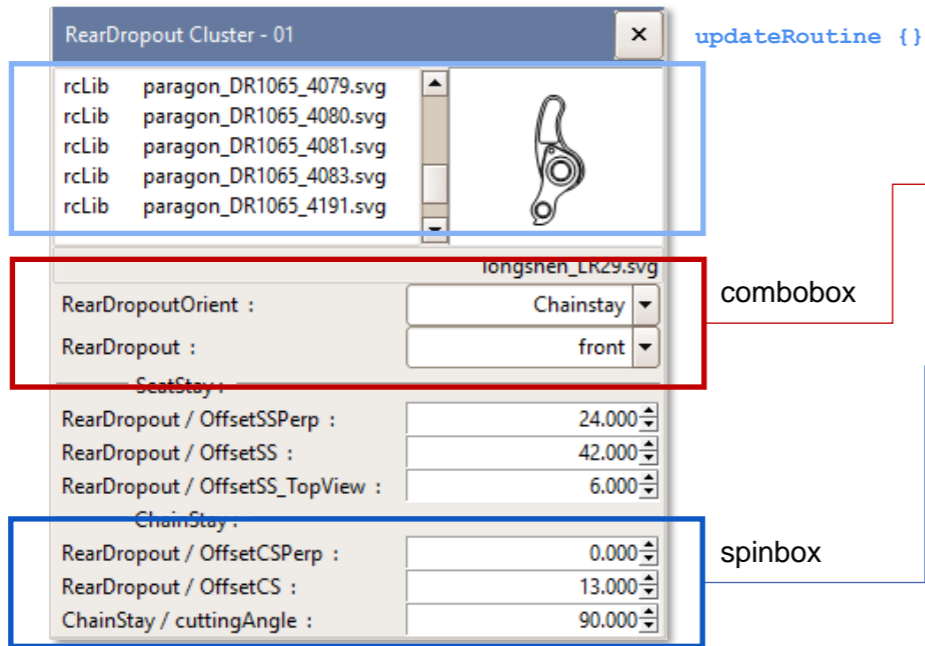


canvas / tkpath

combobox

spinbox

Types of View-Widgets View Widgets in myGUI::viewEdit and update



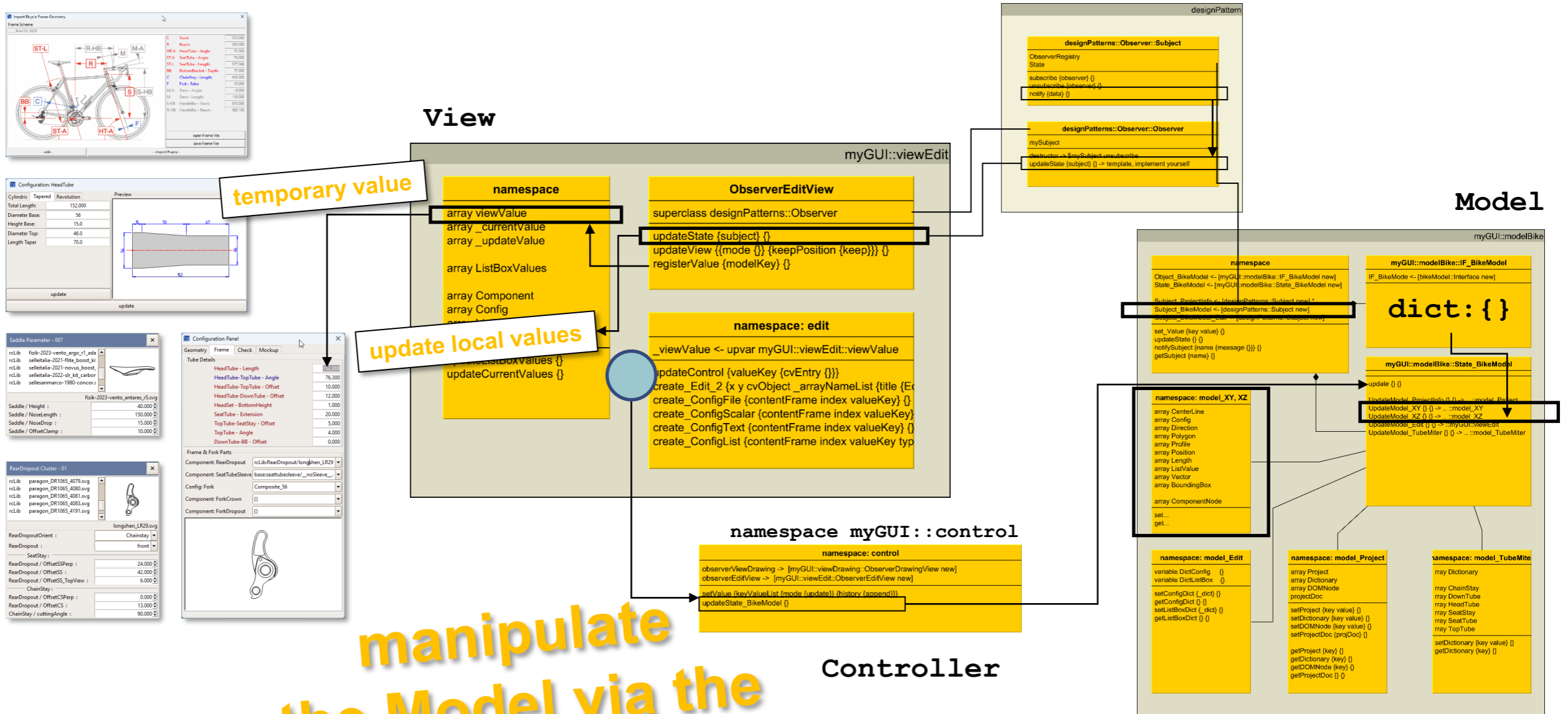
binding: `-textvariable`

Views to
manipulate
the model

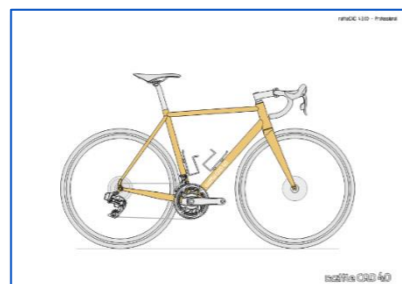
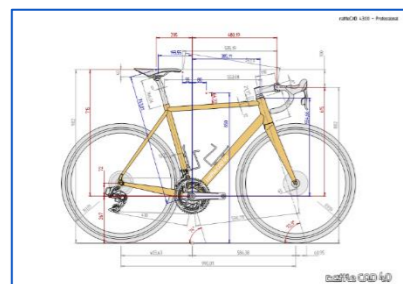
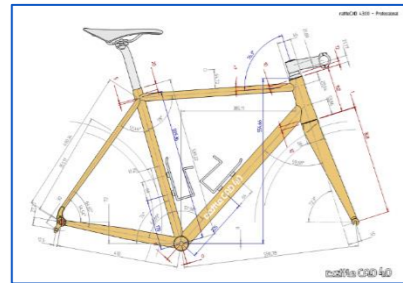
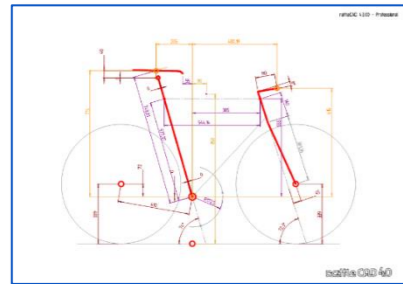
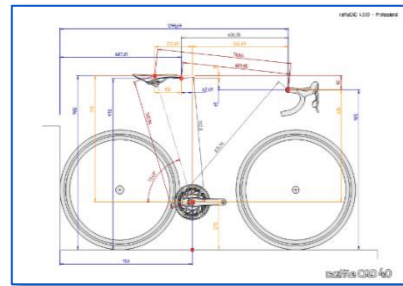
```
proc myGUI::viewEdit::edit::create_ConfigXXXX {contentFrame index valueKey} {  
    #  
    $::myGUI::observerEditView registerValue $valueKey  
    #  
    set varName [format {%s::%s(%s)} [namespace parent] viewValue $valueKey]  
    #  
    #  
    set cfgWidget [ttk::combobox $contentFrame.cb_${index} \  
        -textvariable $varName  
        -values $listBoxContent \  
        -height 10 \  
        -justify right]  
    #  
    #  
    set cfgWidget [spinbox $contentFrame.value_${index} \  
        -textvariable $varName  
        -increment 1 \  
        -justify right \  
        -relief sunken \  
        -bd 1]  
    #  
    set cfgWidget [ttk::entry $contentFrame.value_${index} \  
        -textvariable $varName \  
        -justify right ]  
    #  
    #  
    bind $cfgWidget <<Increment>> [list [namespace current]::bind_SpinBoxButton ...]  
    bind $cfgWidget <<Decrement>> [list [namespace current]::bind_SpinBoxButton ...]  
    bind $cfgWidget <MouseWheel> [list [namespace current]::bind_SpinBoxMouseWheel ...]  
    bind $cfgWidget <Return> [list [namespace current]::updateControl ...]  
    bind $cfgWidget <Tab> [list [namespace current]::updateControl ...]  
    bind $cfgWidget <Double-1> [list [namespace current]::updateControl ...]  
    #  
    bind $cfgWidget <Double-Button-1> {} ;# disable TEntry double-click  
    bind $cfgWidget <Triple-Button-1> {} ;# disable TEntry triple-click  
    #  
    return $cfgWidget  
    #  
}
```

dynamic store for
temporary values

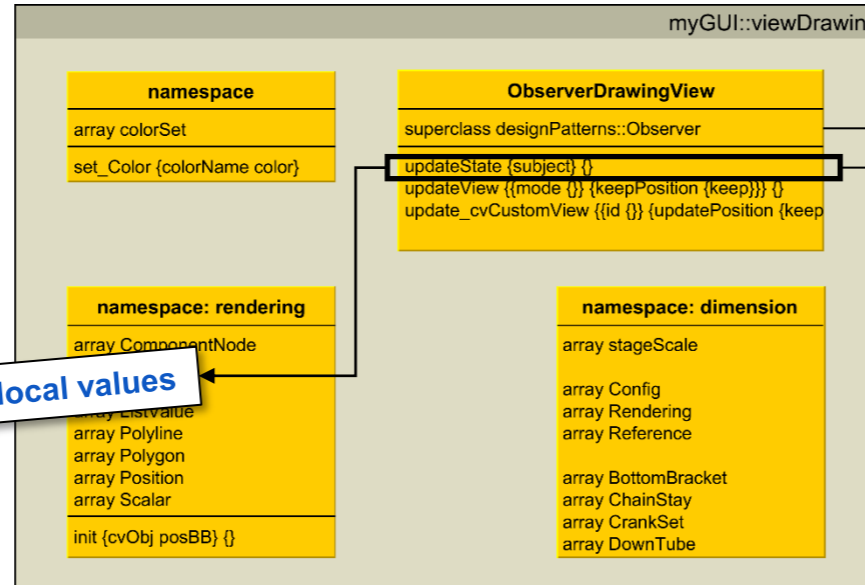
Implementation in Detail: Context – viewEdit (manipulate the model)



Implementation in Detail: Context – Drawing XY, XZ (visualize data only)

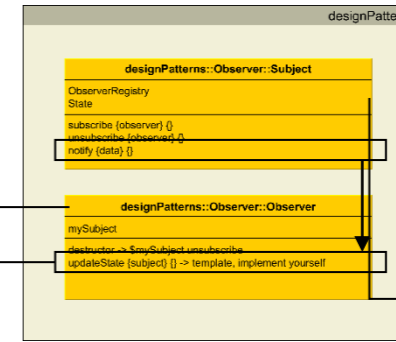


View

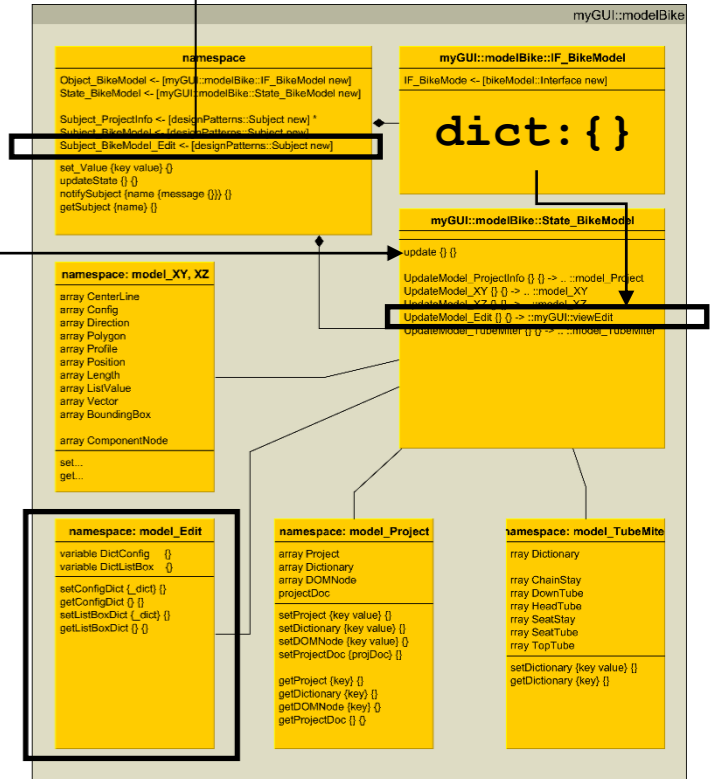


update local values

Drawing views do not know about the MVC-Controller



Model



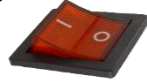
Review & Summary



Results

Synchronize Views

- working



Accelerate the Start-up Process

- stable and working



Software Architecture

- documentation & overview
 - clearer structure
 - Independent packages



Improved testability

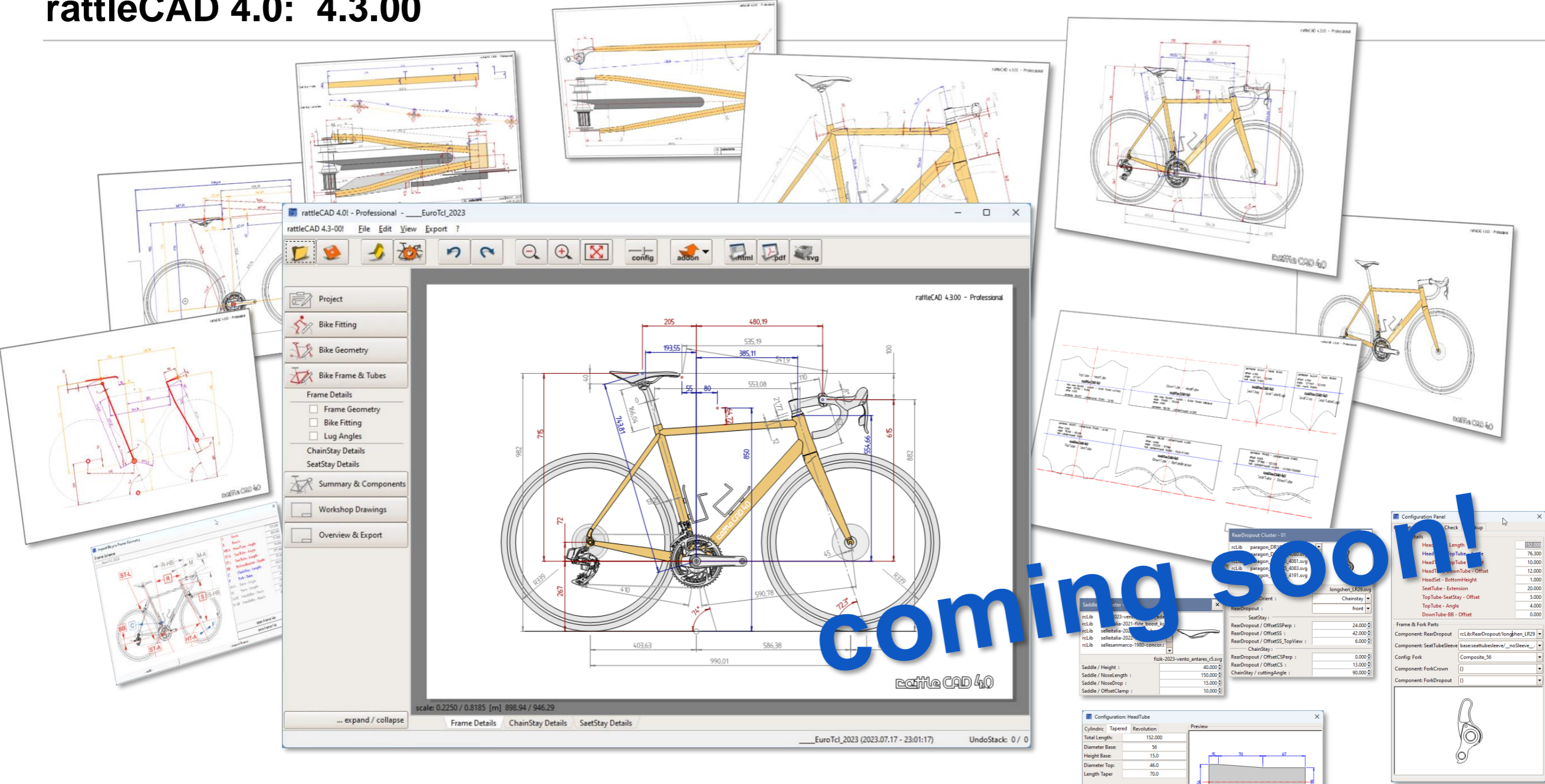
- modular testing
 - e.g: run the GUI without the domain model



Where do I get the Observer-Class libraries?

- Download:
 - <https://github.com/level44/design-patterns-in-tcl>
- Create yourself:
 - https://sourcemaking.com/design_patterns

rattleCAD 4.0: 4.3.00



**Thanks for all of the support
from the Tcl/Tk community!**



Manfred ROSENBERGER