Graz University of Technology

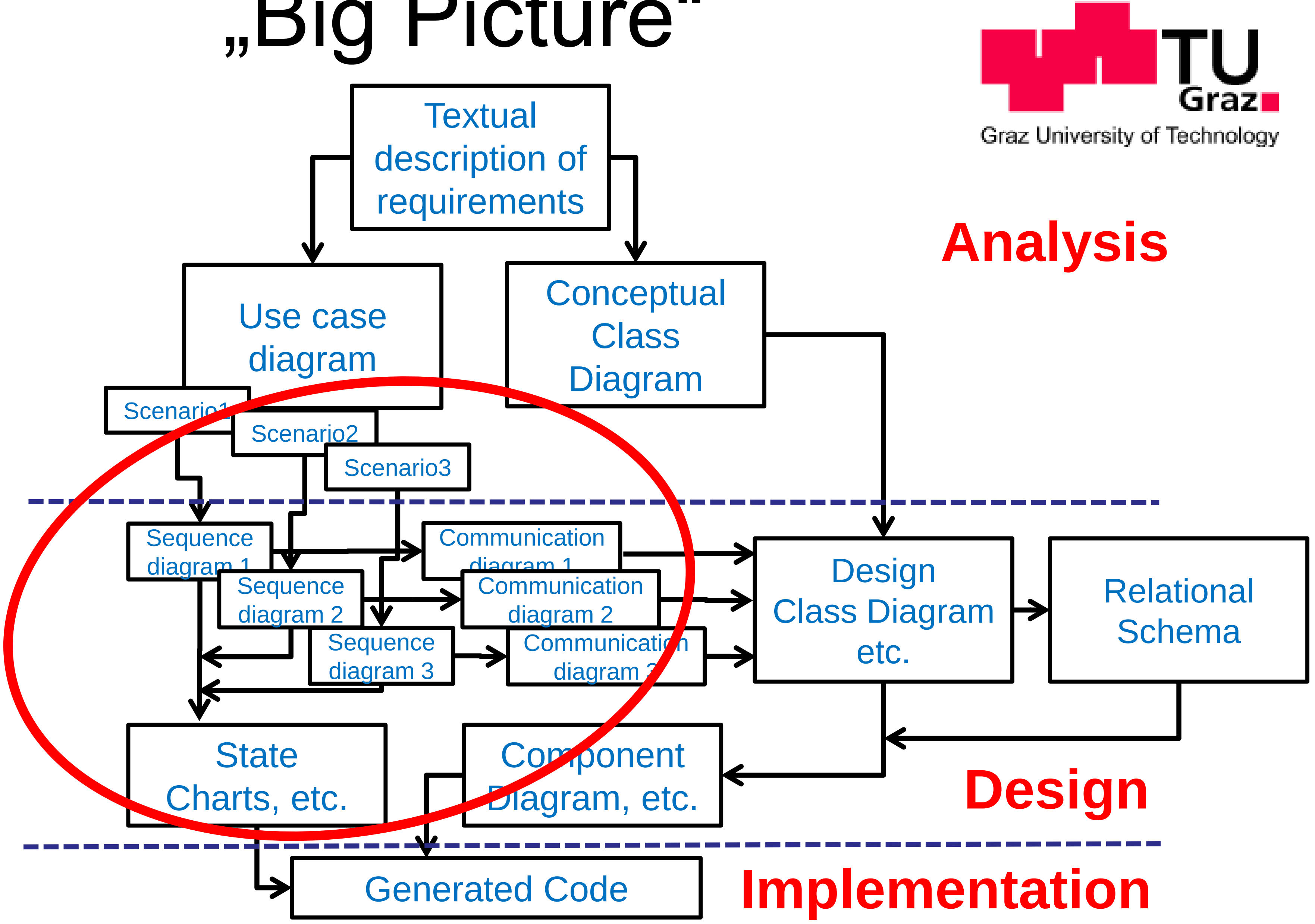
Object-Oriented Analysis & Design (OAD)

Sequence Diagrams & State Charts

https://youtu.be/lsqeS8hOtIc

Alexander Felfernig
Institute for Software Technology
Inffeldgasse 16b/2

„Big Picture“

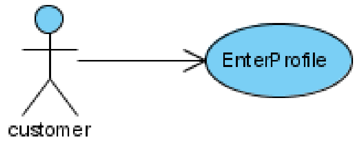


TU Graz
Graz University of Technology
Textual description of requirements
Use case diagram
Conceptual Class Diagram
Analysis
Scenario1
Scenario2
Scenario3
Sequence diagram 1
Sequence diagram 2
Sequence diagram 3
Communication diagram 1
Communication diagram 2
Communication diagram 3
Design Class Diagram etc.
Relational Schema
State Charts, etc.
Component Diagram, etc.
Design
Generated Code
Implementation

Use Cases Revisited

Graz University of Technology

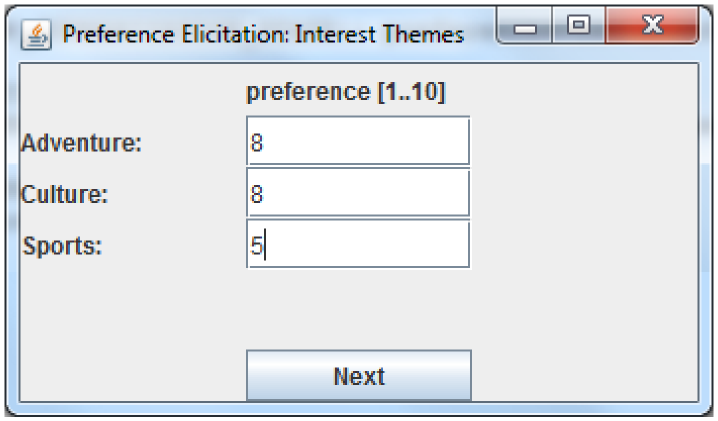
- use cases are exploited **to scope the functionality** of the system



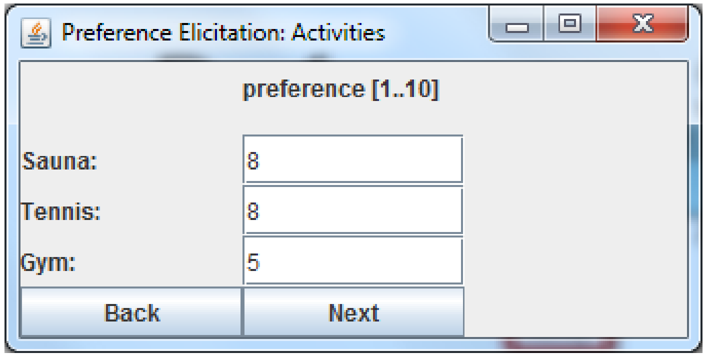
- they define **which functions will be included**
- use cases have an underlying **business goal**
- **disadvantage:** use cases cannot express event dependencies & alternatives very well
- **sequence diagrams** can do this …
- **examples**: preference elicitation, watch interface

Preference Elicitation: Use Case

1. the **user** activates “Preference Elicitation”
2. the **system** asks for preferences regarding **interest themes**
3. the **user** enters his/her preferences
4. the user presses the “next” button
5. the **system** asks for preferences regarding **activities**
6. the **user** enters his/her preferences
7. the **system** displays a complete summary of the user input
8. the **user** confirms the preferences with pressing the “store button”


Preference Elicitation: Interest Themes

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5

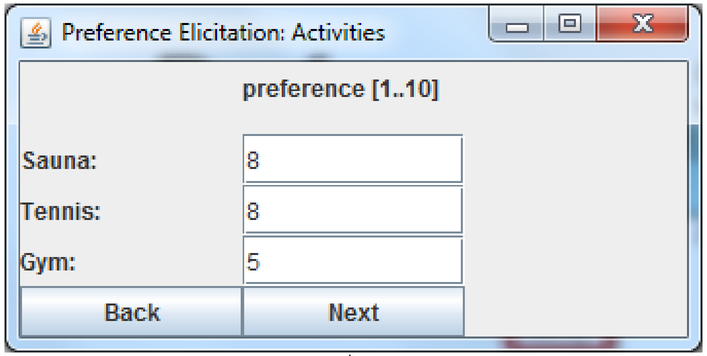

Preference Elicitation: Activities

	preference [1..10]
Sauna:	8
Tennis:	8
Gym:	5

Preference Elicitation: Use Case

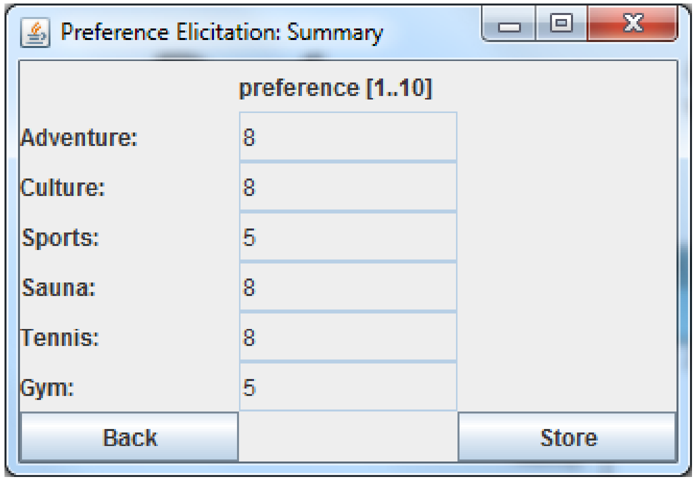
Graz University of Technology

1. the **user** activates “Preference Elicitation”
2. the **system** asks for preferences regarding interest themes
3. the **user** enters his/her preferences
4. the user presses the “next” button
5. the **system** asks for preferences regarding activities
6. the **user** enters his/her preferences
7. the **system** displays a complete summary of the user input
8. the **user** confirms the preferences with pressing the “store button”



Preference Elicitation: Activities

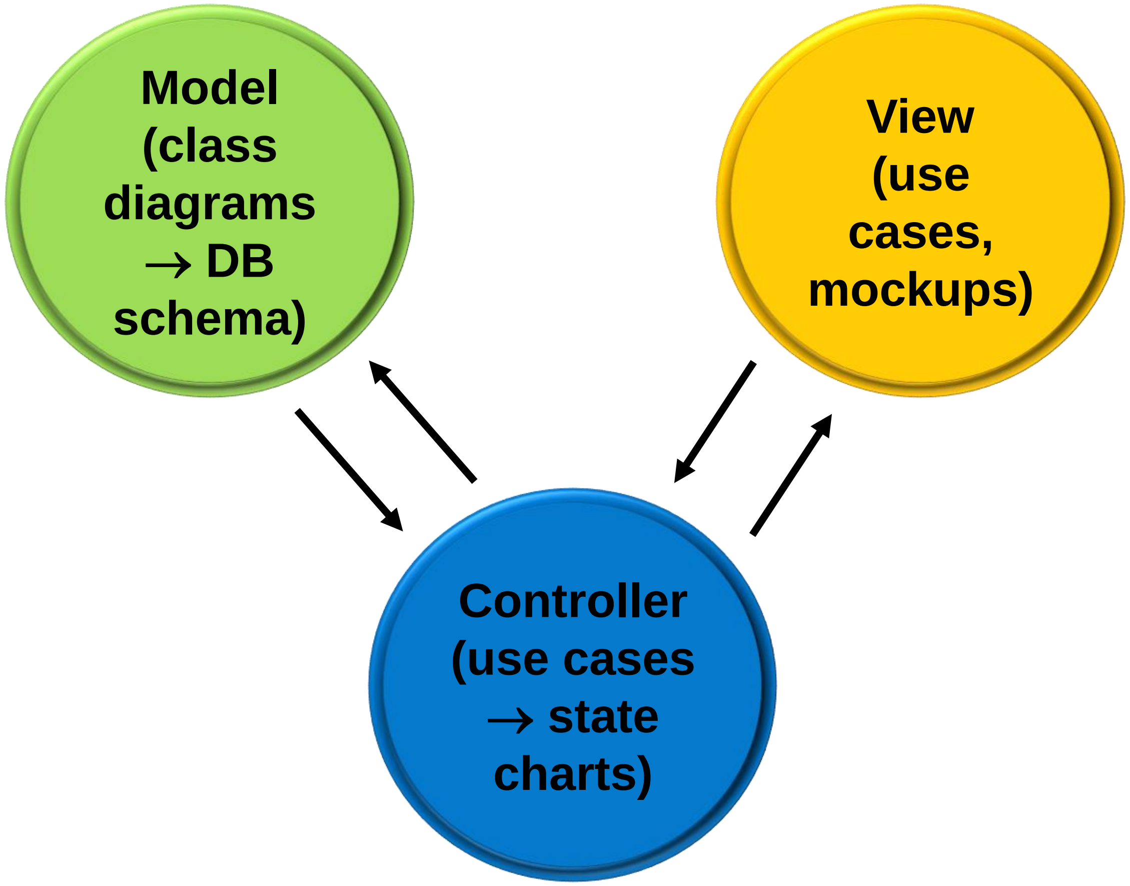
	preference [1..10]
Sauna:	8
Tennis:	8
Gym:	5



Preference Elicitation: Summary

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Model View Controller

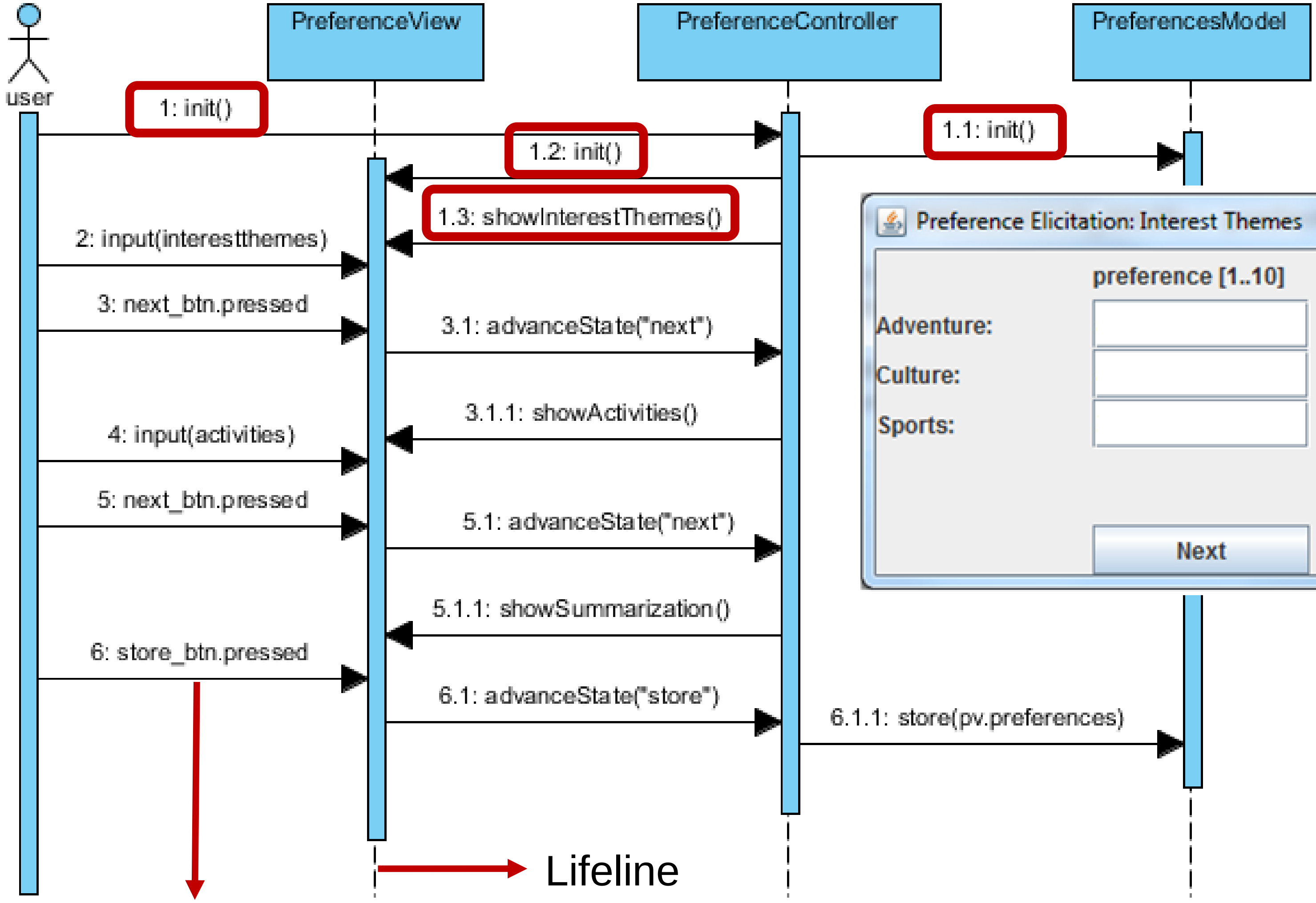

Model (class diagrams → DB schema)
View (use cases, mockups)
Controller (use cases → state charts)

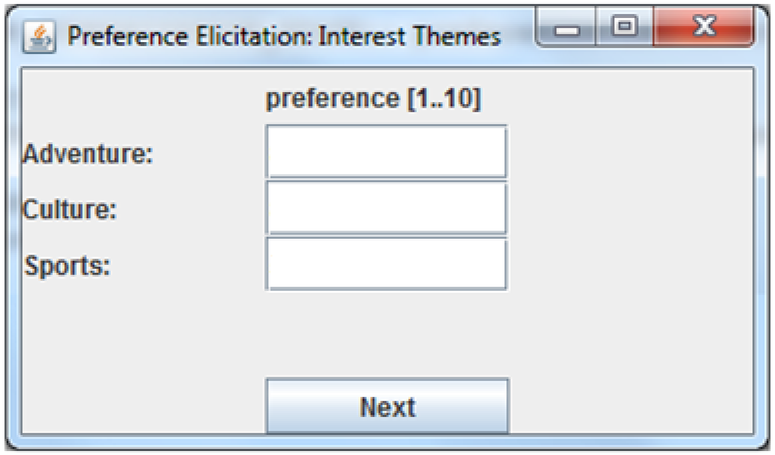
Model View Controller

- No direct coupling of non-UI and UI objects
  - e.g., no reference of a domain (**model**) object to a Java JFrame object
  - ☞ non-windowing objects may be used by other applications as well
- No application logic in the UI object
  - UI objects (**view**) for initializing UI elements, receiving UI events (e.g., button clicks)
  - Delegation of requests for application logic to non-UI objects (e.g., a **controller**)

Sequence Diagram

Graz University of Technology




Preference Elicitation: Interest Themes

preference [1..10]

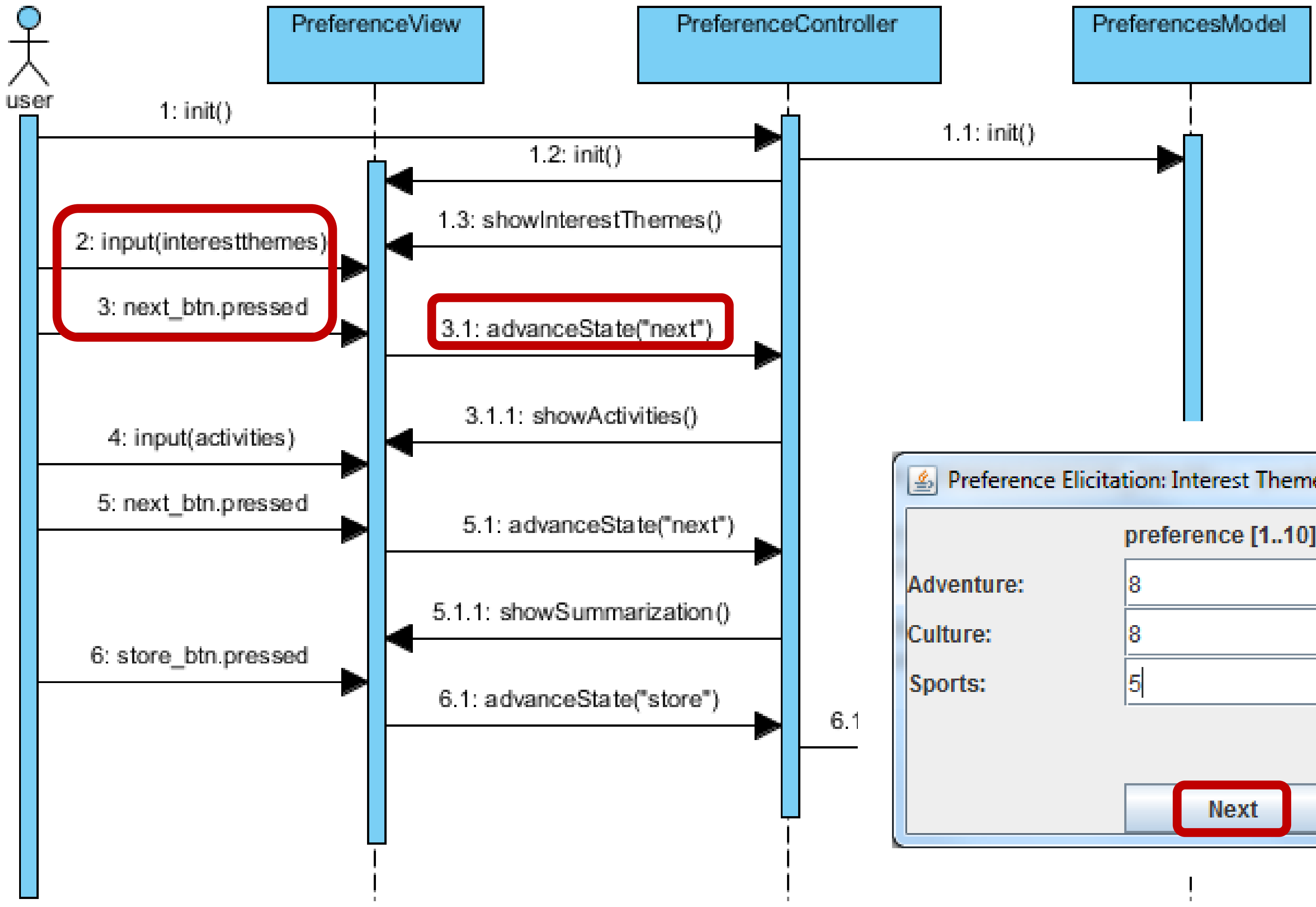
Adventure:	
Culture:	
Sports:	

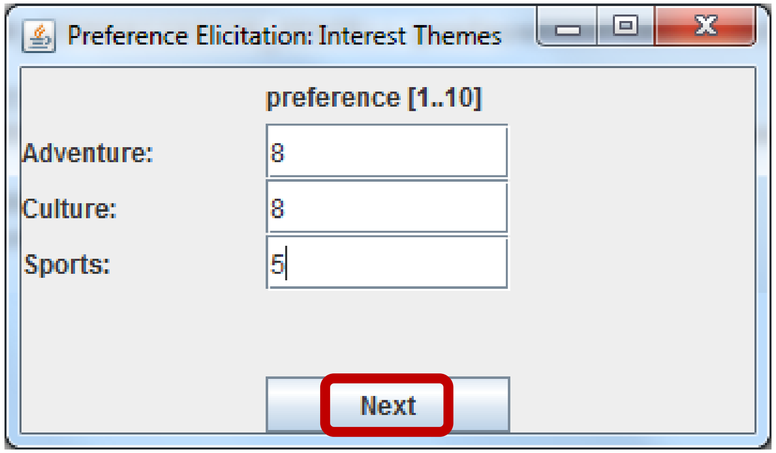
Lifeline

Synchronous message

Sequence Diagram

Graz University of Technology




Preference Elicitation: Interest Themes

preference [1..10]

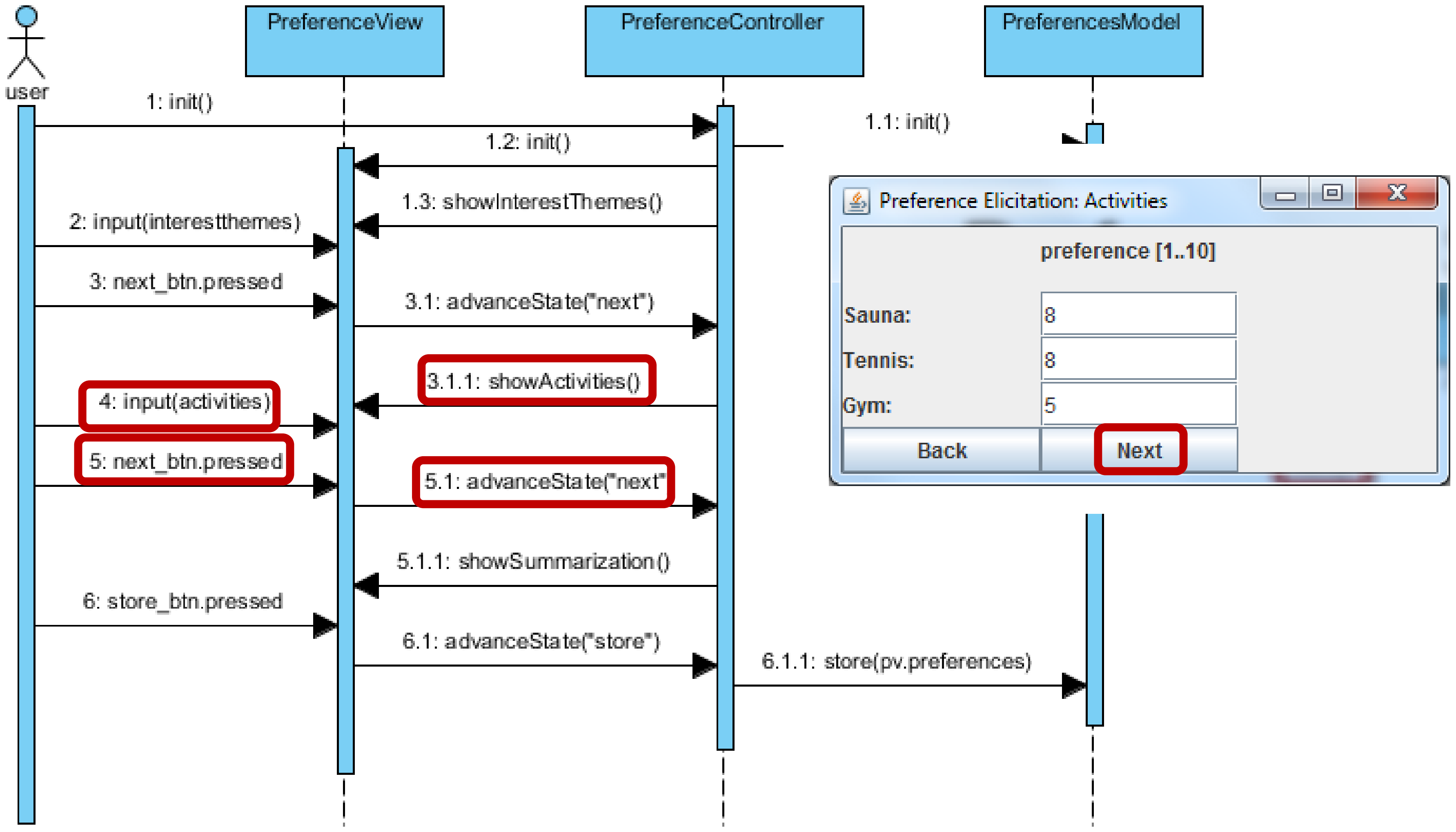
Adventure: 8

Culture: 8

Sports: 5

Sequence Diagram

Graz University of Technology



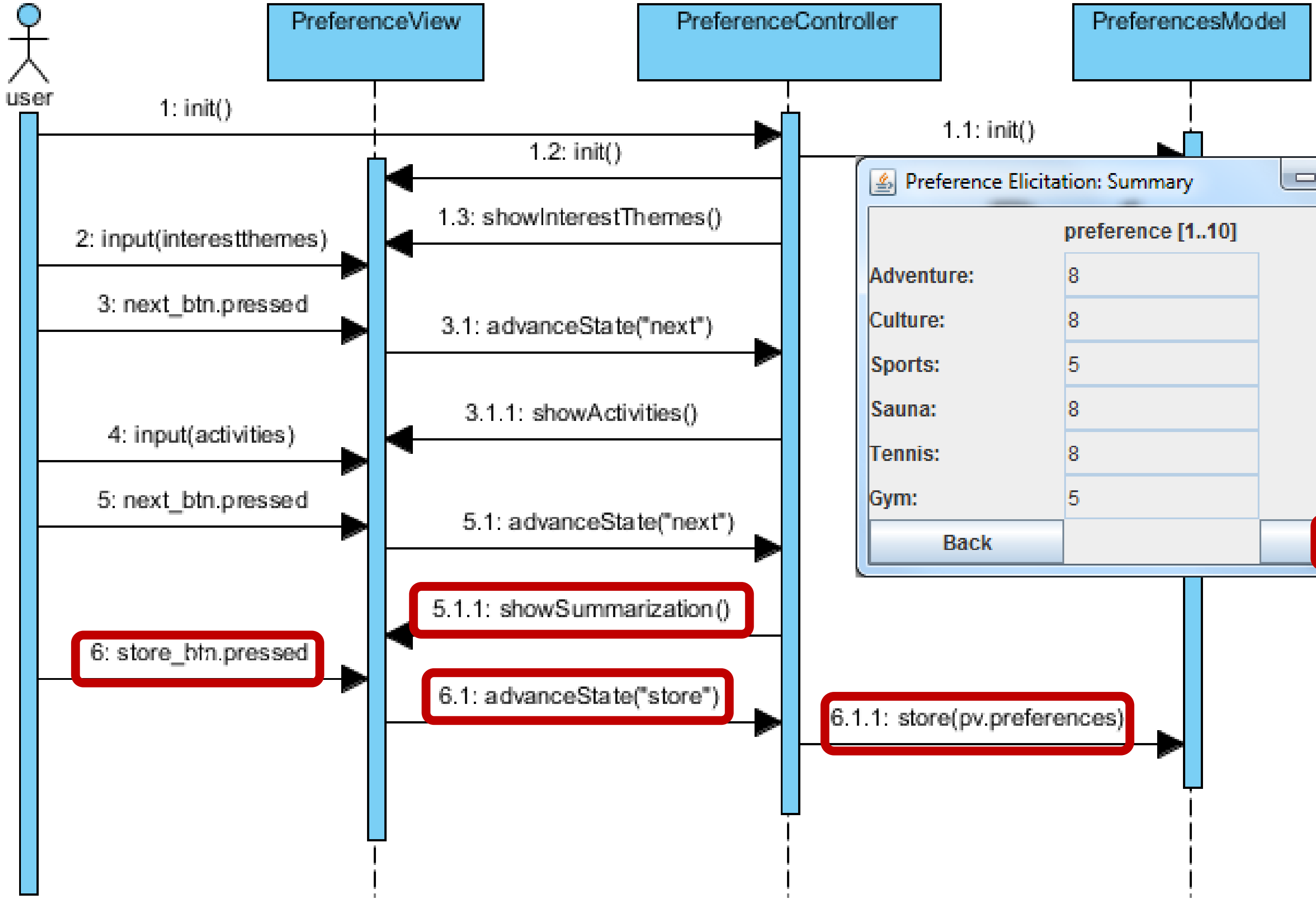
Preference Elicitation: Activities

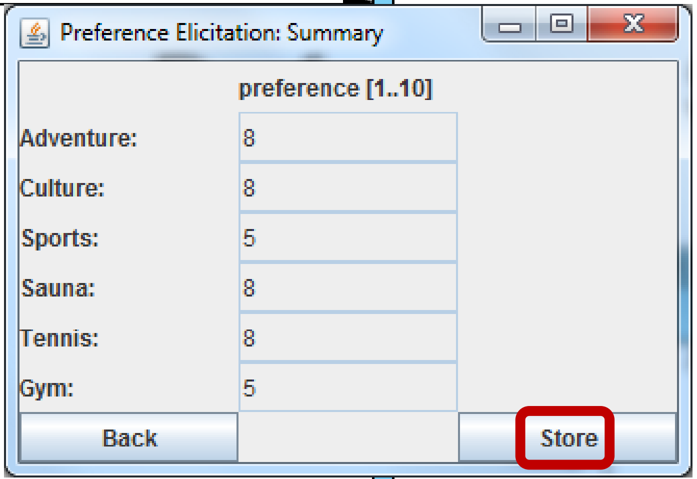
preference [1..10]

Sauna: 8
Tennis: 8
Gym: 5

Sequence Diagram

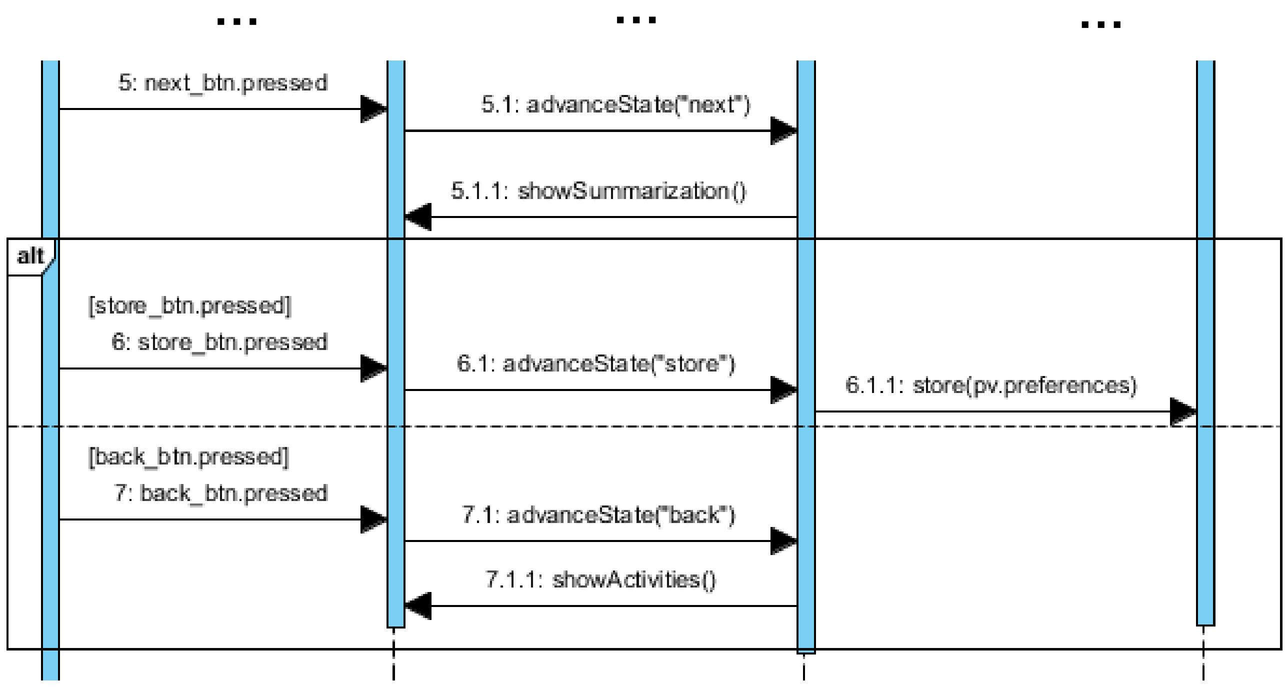
Graz University of Technology





	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Preference Elicitation: Alternative Fragments


...
...
...
5: next_btn.pressed
5.1: advanceState("next")
5.1.1: showSummarization()
alt
[store_btn.pressed]
6: store_btn.pressed
6.1: advanceState("store")
6.1.1: store(pv.preferences)
[back_btn.pressed]
7: back_btn.pressed
7.1: advanceState("back")
7.1.1: showActivities()

break

if fragment is executed, the remainder of the sequence is ignored

opt

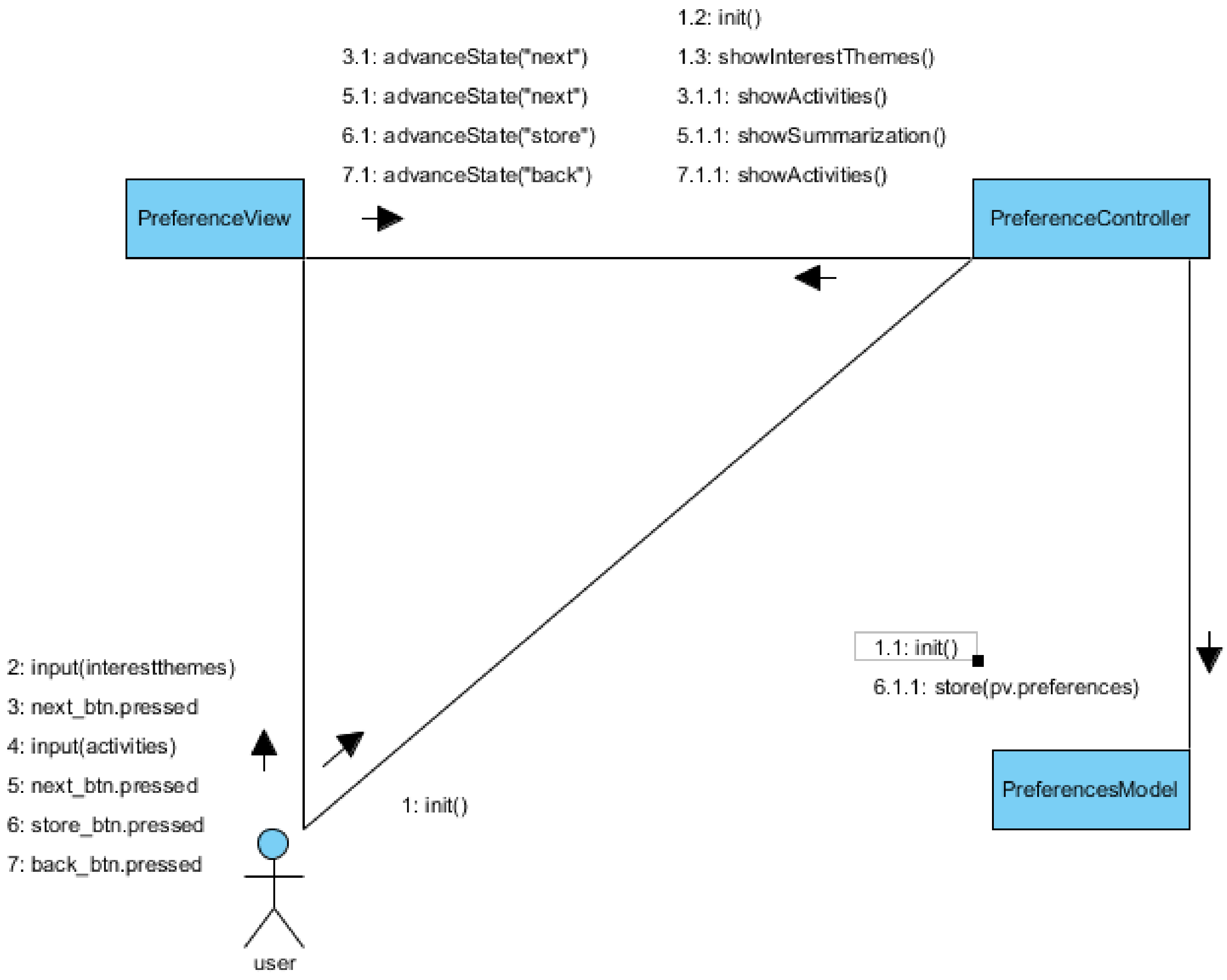
optional block, executed if condition is true

loop

as long as loop condition fulfilled, execute sequence block

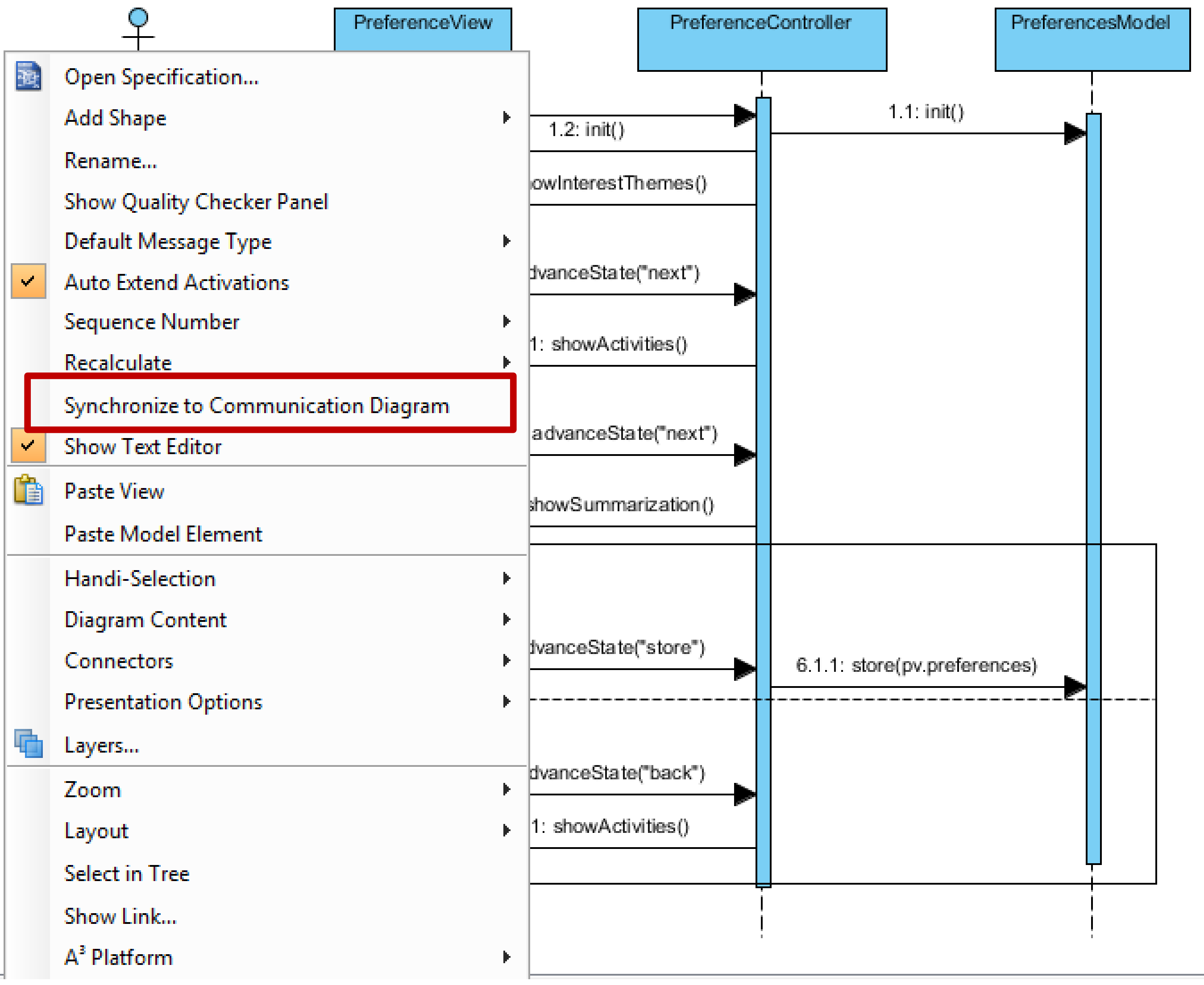
See also: https://www.uml-diagrams.org/sequence-diagrams-combined-fragment.html

Preference Elicitation: Collaboration Diagram



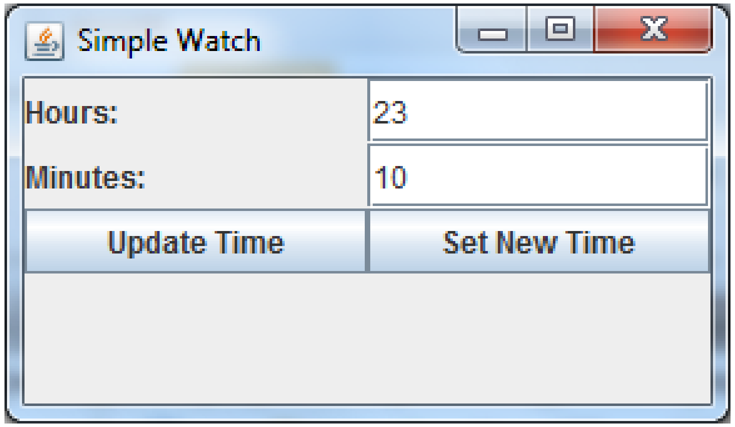
Generation in Visual Paradigm

Graz University of Technology

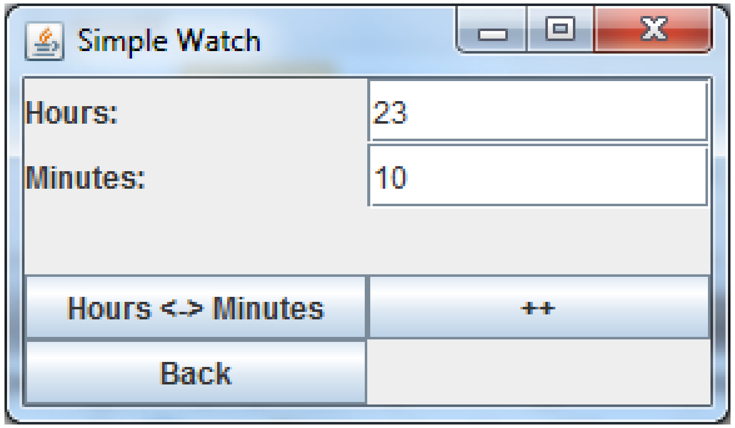


Simple Watch: Use Case

1. the **user** activates “Update Time”
2. the **system** displays the current time
3. the **user** enters “Set New Time”
4. the **system** adapts the set of available buttons
5. the **user** presses the “++” button
6. the **system** increments the hour information
7. the **user** presses the “Back” button
8. the **system** adapts the set of available buttons

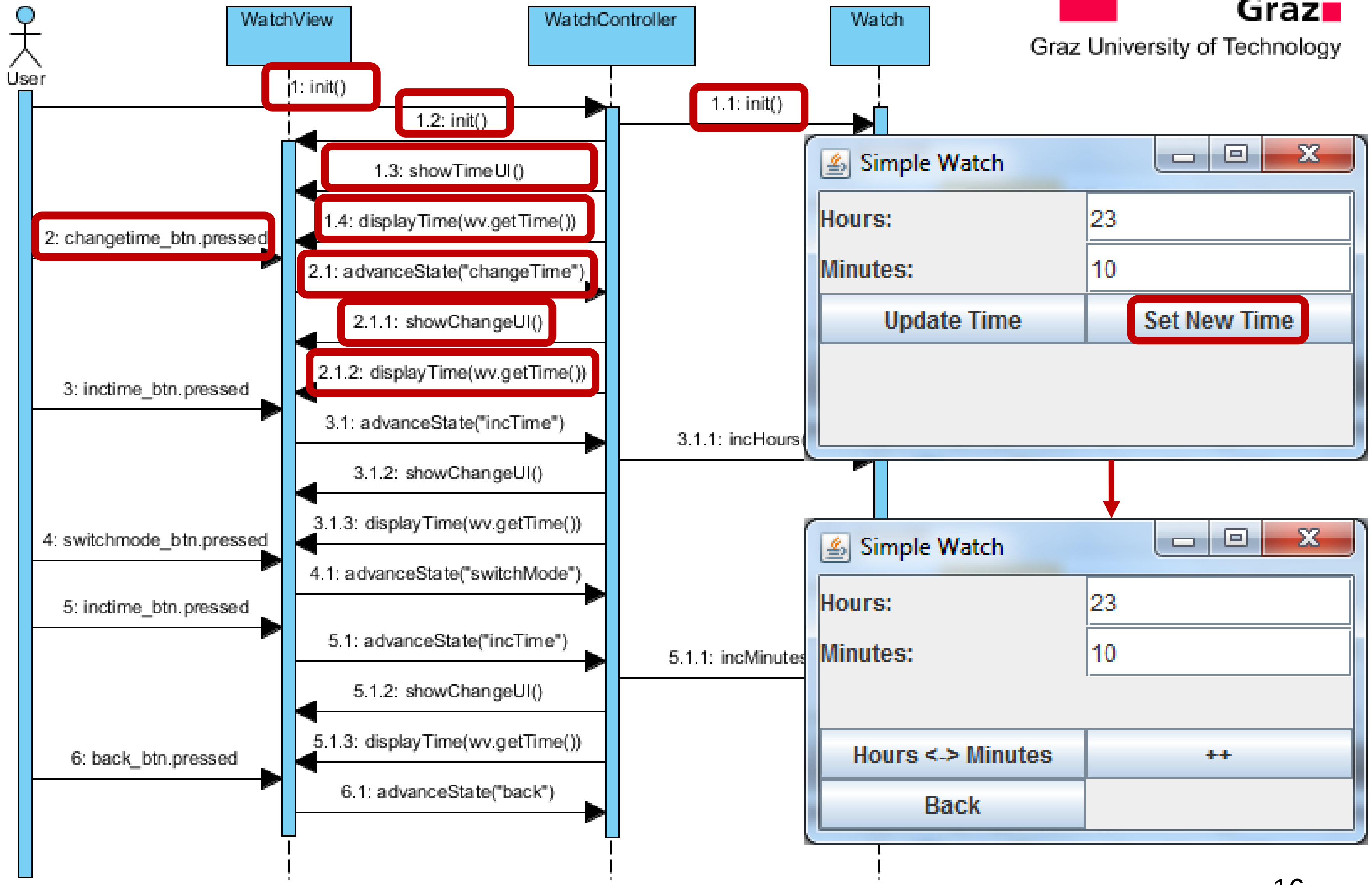


Hours: 23
Minutes: 10



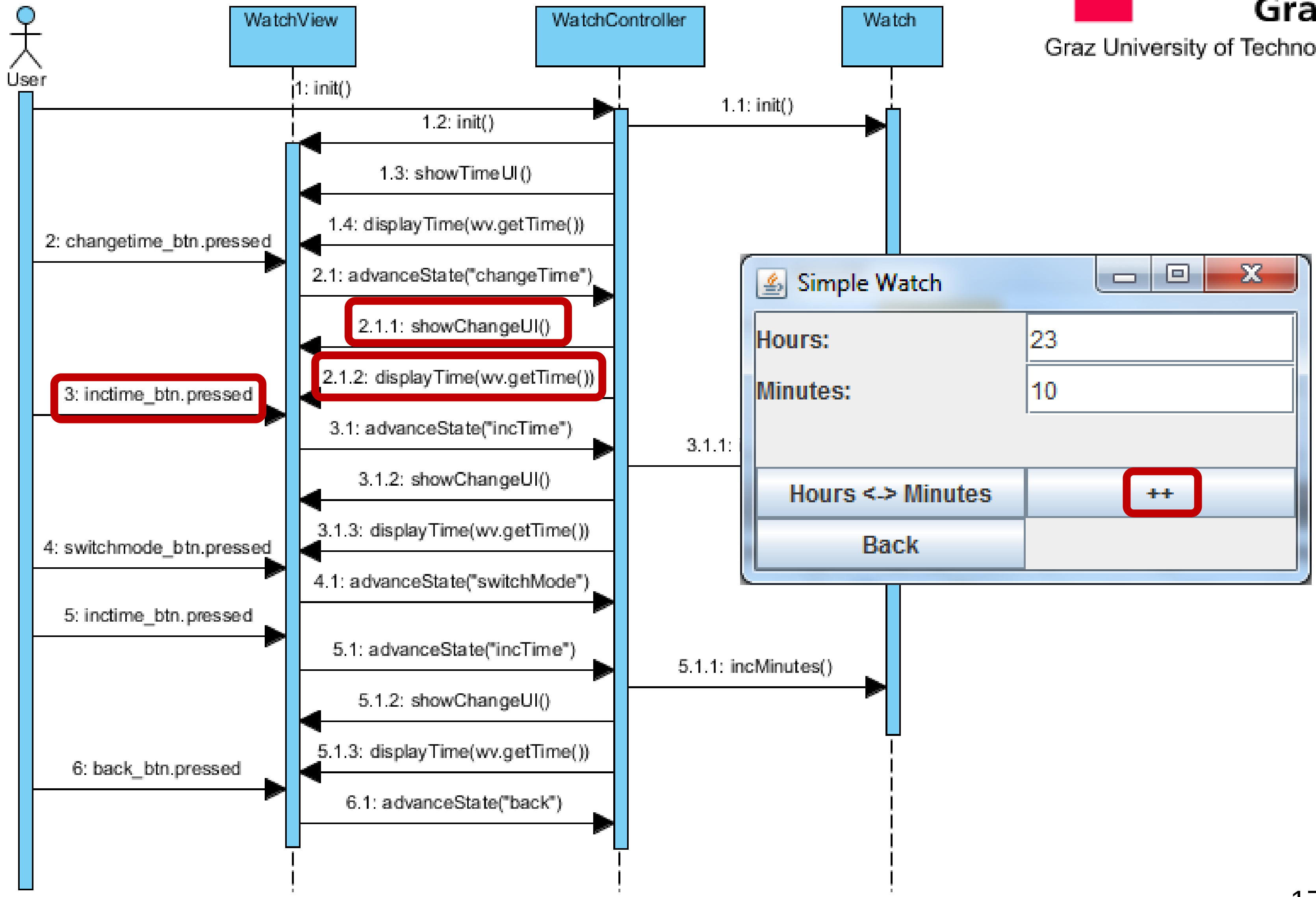
Hours: 23
Minutes: 10

Sequence Diagram



Sequence Diagram

Graz University of Technology

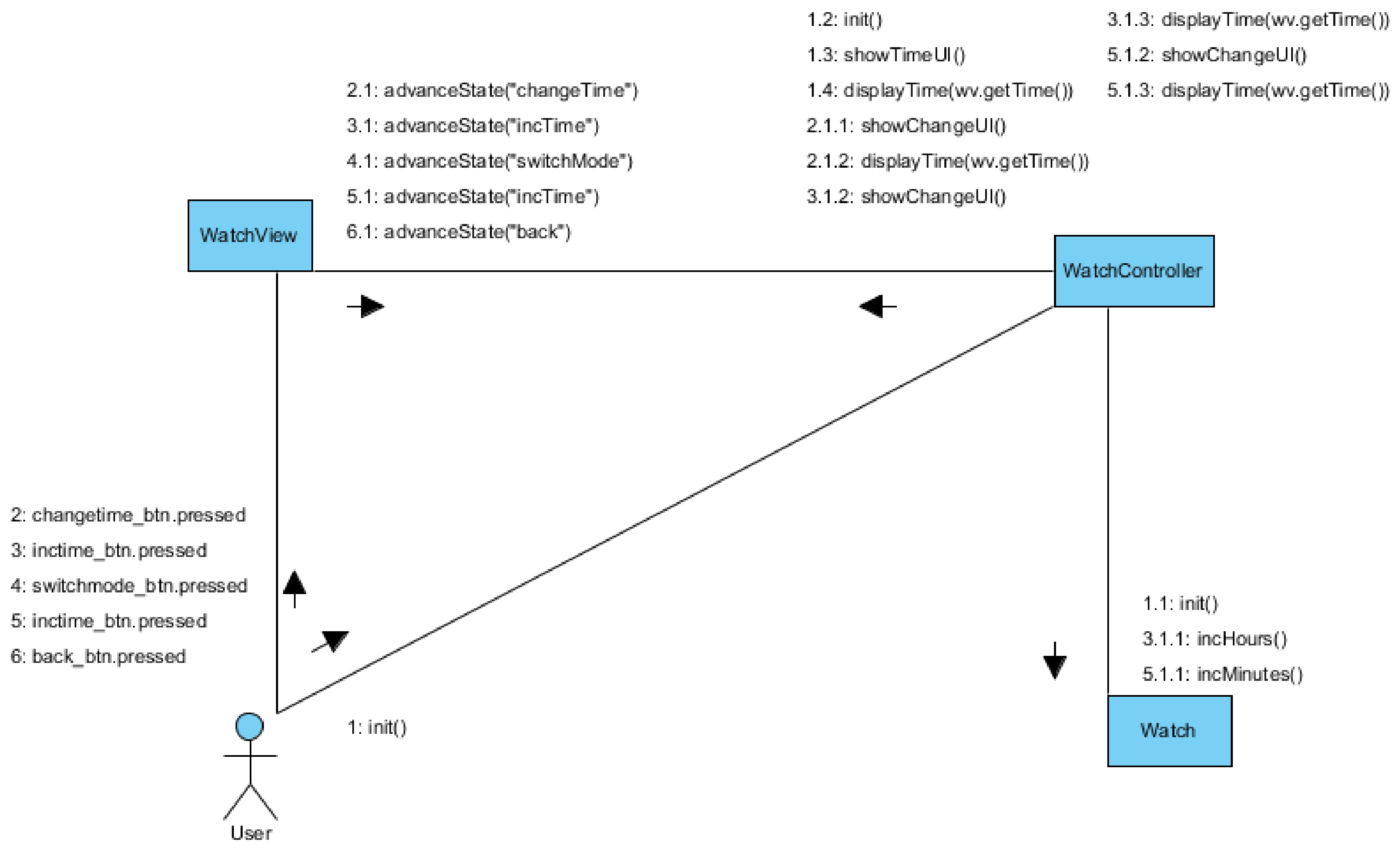


User
WatchView
WatchController
Watch
1: init()
1.1: init()
1.2: init()
1.3: showTimeUI()
1.4: displayTime(wv.getTime())
2: changetime_btn.pressed
2.1: advanceState("changeTime")
2.1.1: showChangeUI()
2.1.2: displayTime(wv.getTime())
3: inctime_btn.pressed
3.1: advanceState("incTime")
3.1.1:
3.1.2: showChangeUI()
3.1.3: displayTime(wv.getTime())
4: switchmode_btn.pressed
4.1: advanceState("switchMode")
5: inctime_btn.pressed
5.1: advanceState("incTime")
5.1.1: incMinutes()
5.1.2: showChangeUI()
5.1.3: displayTime(wv.getTime())
6: back_btn.pressed
6.1: advanceState("back")
Simple Watch
Hours:
23
Minutes:
10
Hours <-> Minutes
++
Back

. . .

Simple Watch: Collaboration Diagram

Graz University of Technology



1.2: init()
1.3: showTimeUI()
1.4: displayTime(wv.getTime())
2.1.1: showChangeUI()
2.1.2: displayTime(wv.getTime())
3.1.2: showChangeUI()
3.1.3: displayTime(wv.getTime())
5.1.2: showChangeUI()
5.1.3: displayTime(wv.getTime())
2.1: advanceState("changeTime")
3.1: advanceState("incTime")
4.1: advanceState("switchMode")
5.1: advanceState("incTime")
6.1: advanceState("back")
WatchView
WatchController
2: changetime_btn.pressed
3: inctime_btn.pressed
4: switchmode_btn.pressed
5: inctime_btn.pressed
6: back_btn.pressed
1.1: init()
3.1.1: incHours()
5.1.1: incMinutes()
1: init()
Watch
User

Collaboration (Communication) Diagrams

- **„Communication Diagrams“** in UML 2.0
- Basically contain the **same information as sequence diagrams**
- Focus on the **connection between objects** and not on the time aspect
- Give a **better overview of object communication**

Component Diagrams

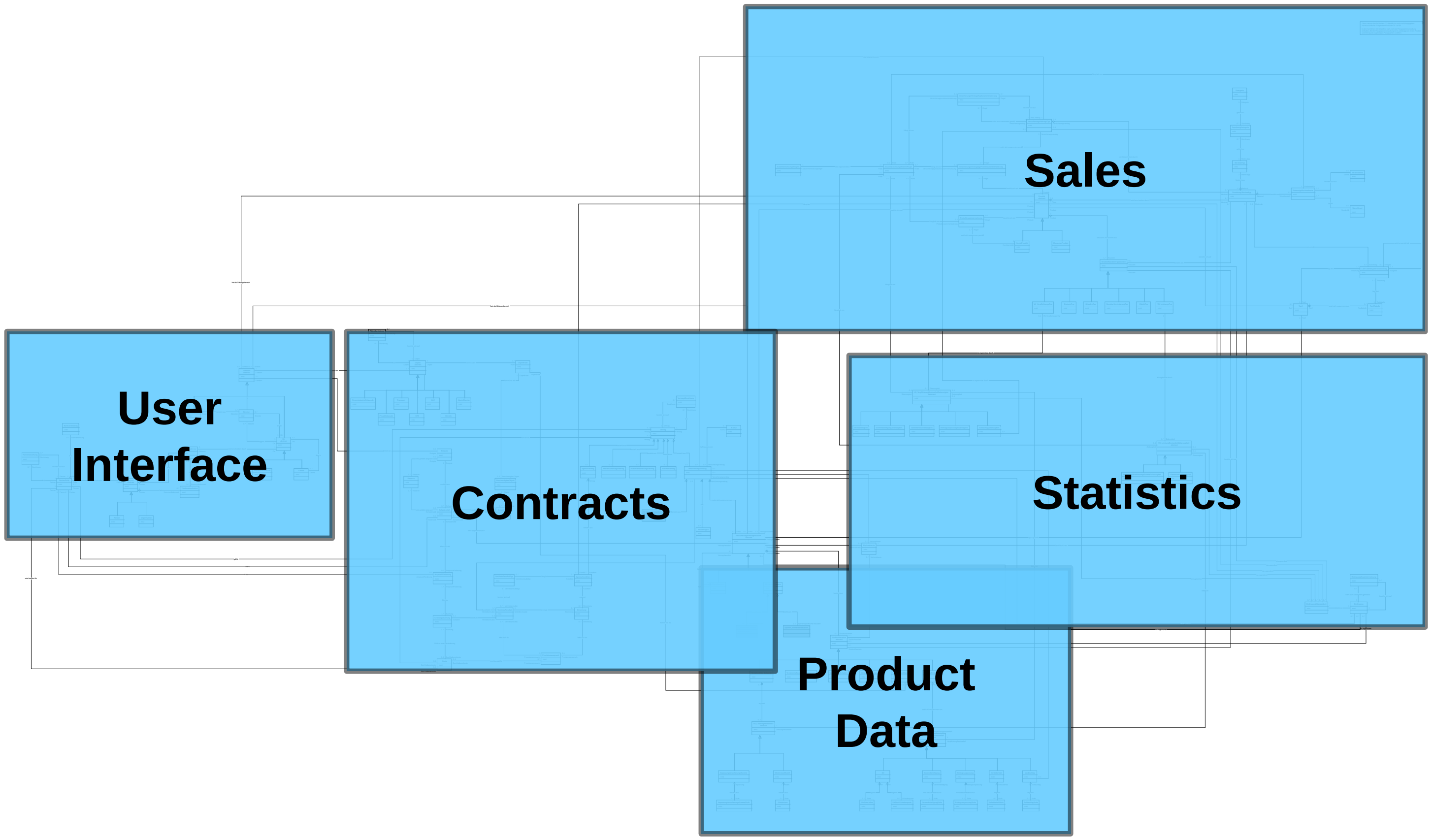
- Class diagrams are an important means needed for modeling **structural properties of the application** (domain)
- **Problem**: clearness/understandability of **large domain models**
- **Solution**:
  - identification of **coherent model parts** (“components”)
  - components are **connected via interfaces**
  - principle of **„divide and conquer“**

Component Diagrams: Components

- **Collection of elements** (classes or other components)
- Elements are **logically coherent**, loosely coupled with elements of other components
- Offer functionalities via **clearly defined interfaces (basic datatypes)**
- **Can be substituted** by other components that offer the same or more functionalities

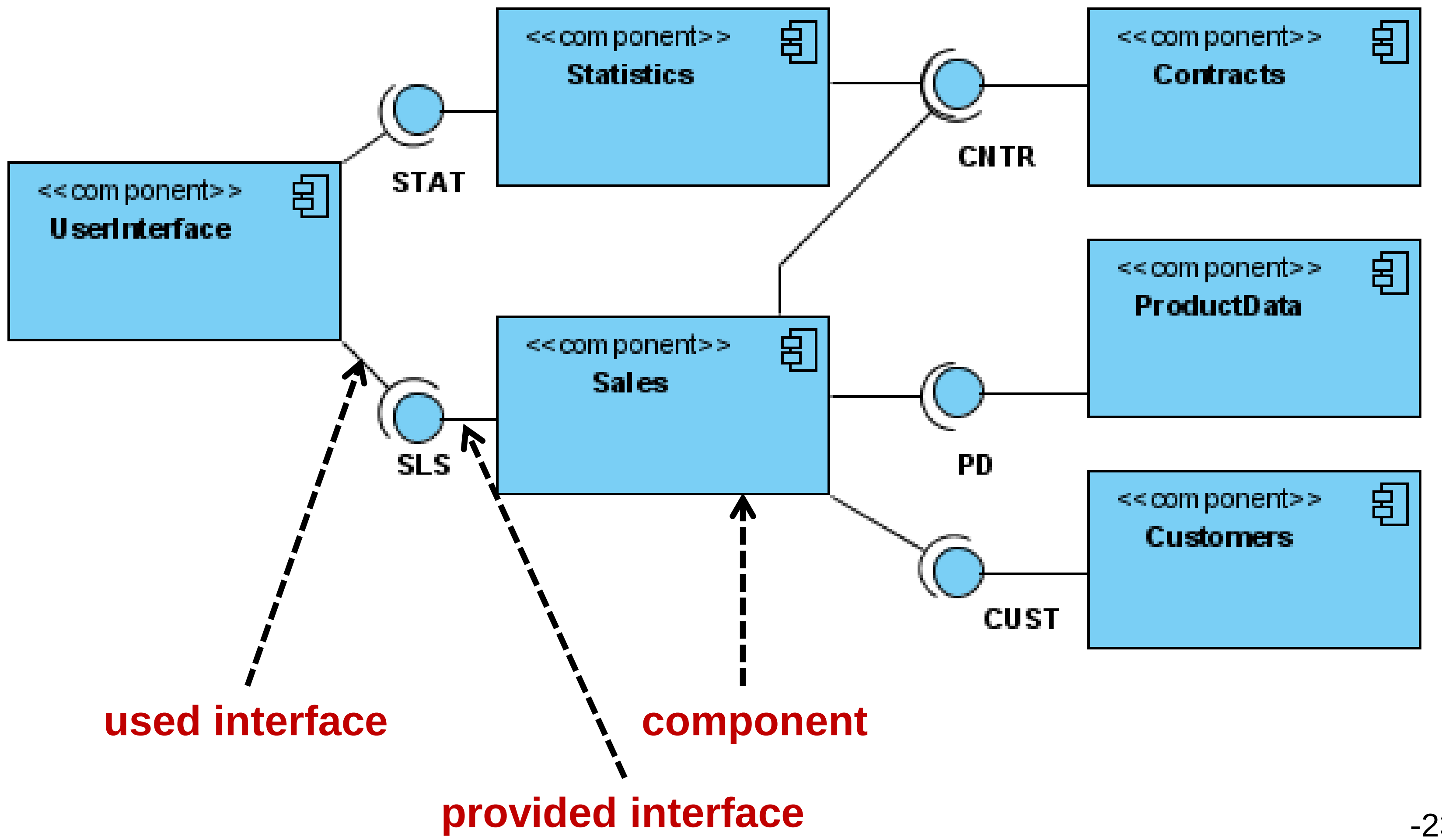
Component Diagrams: CRM System

Graz University of Technology

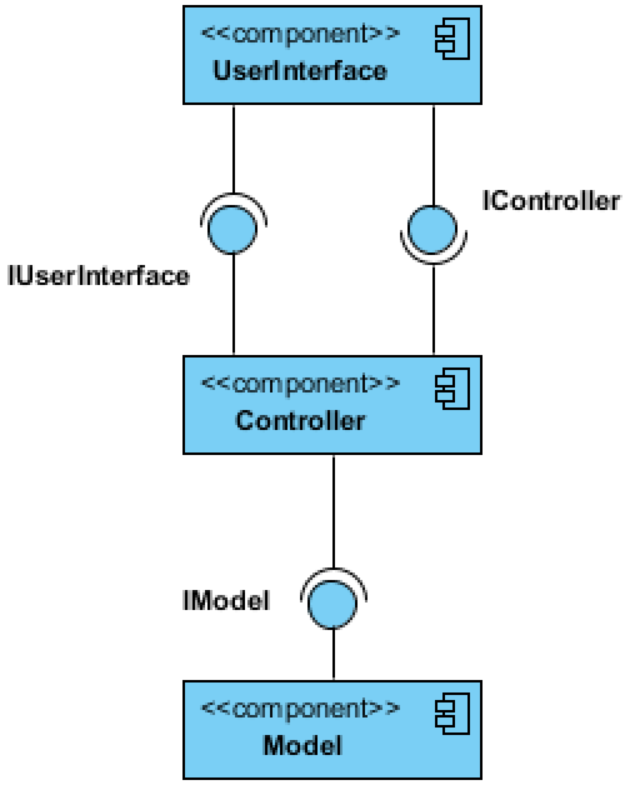


Component Diagrams: UML Representation

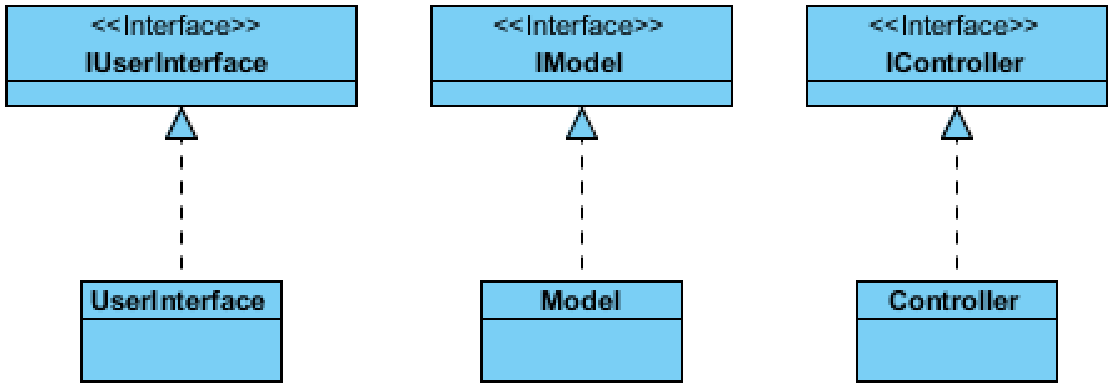
Graz University of Technology



Component Diagrams: Model View Controller



```
public interface IUserInterface {}
public interface IController {}
public interface IModel {}
public class Controller implements IController {}
public class Model implements IModel {}
public class UserInterface implements IUserInterface {}
```

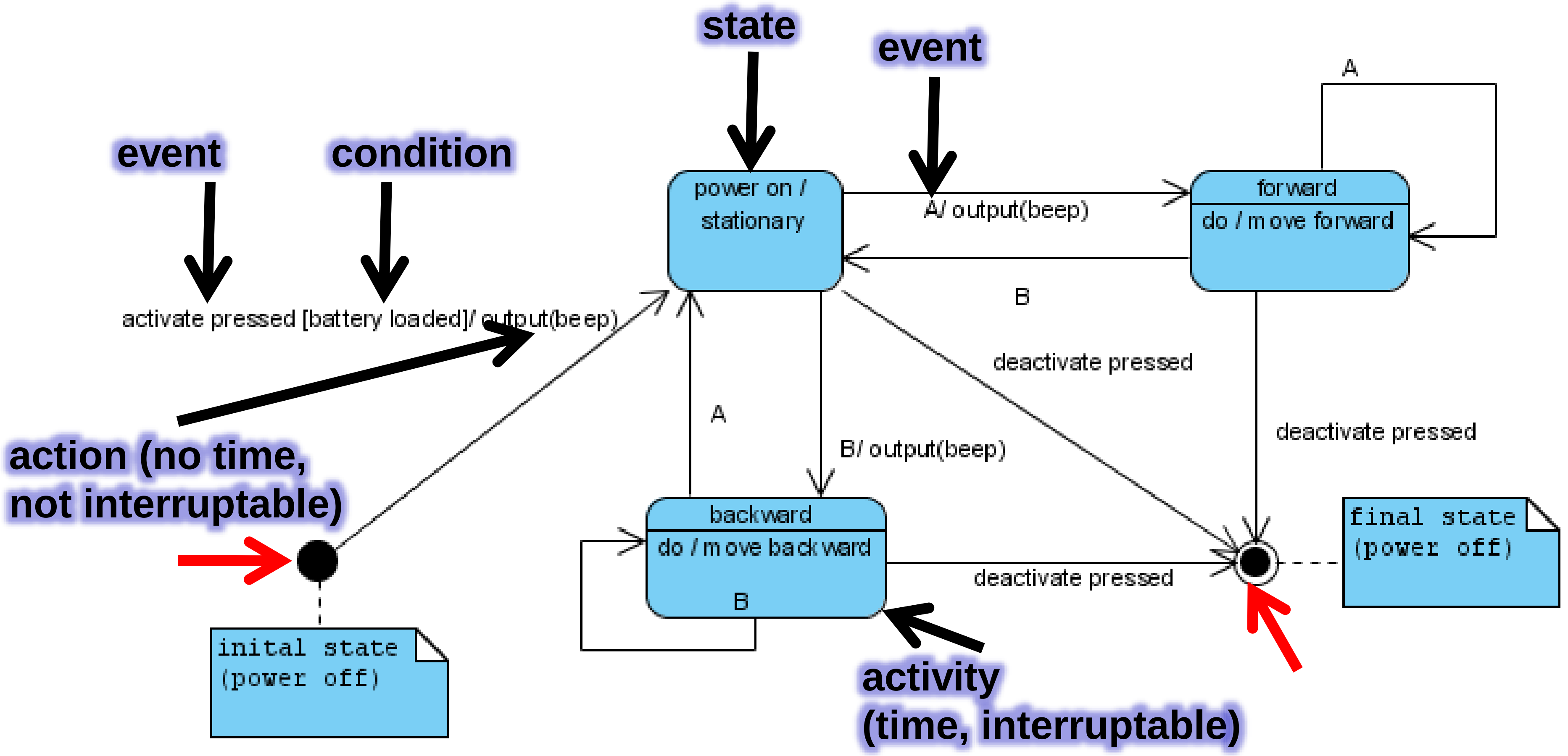


State Charts

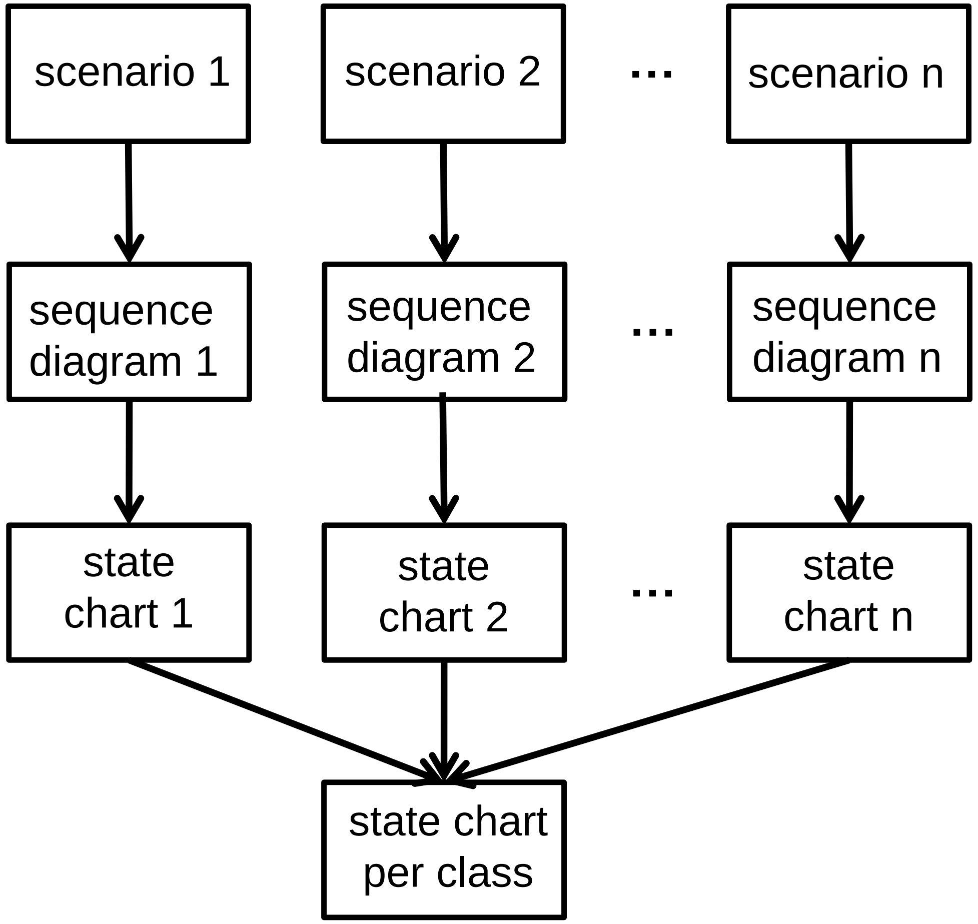
- UML „**State Chart Diagrams**“
- Show the **sequence of states of an object** within the scope of it‘s lifetime
- Specify under which **conditions** changes of the state take place
- **Modeling concepts**: state, event, condition, action, activity, initial state, final state

State Charts: Remote Controlled Toy Car

Graz University of Technology



Process: Deriving State Charts



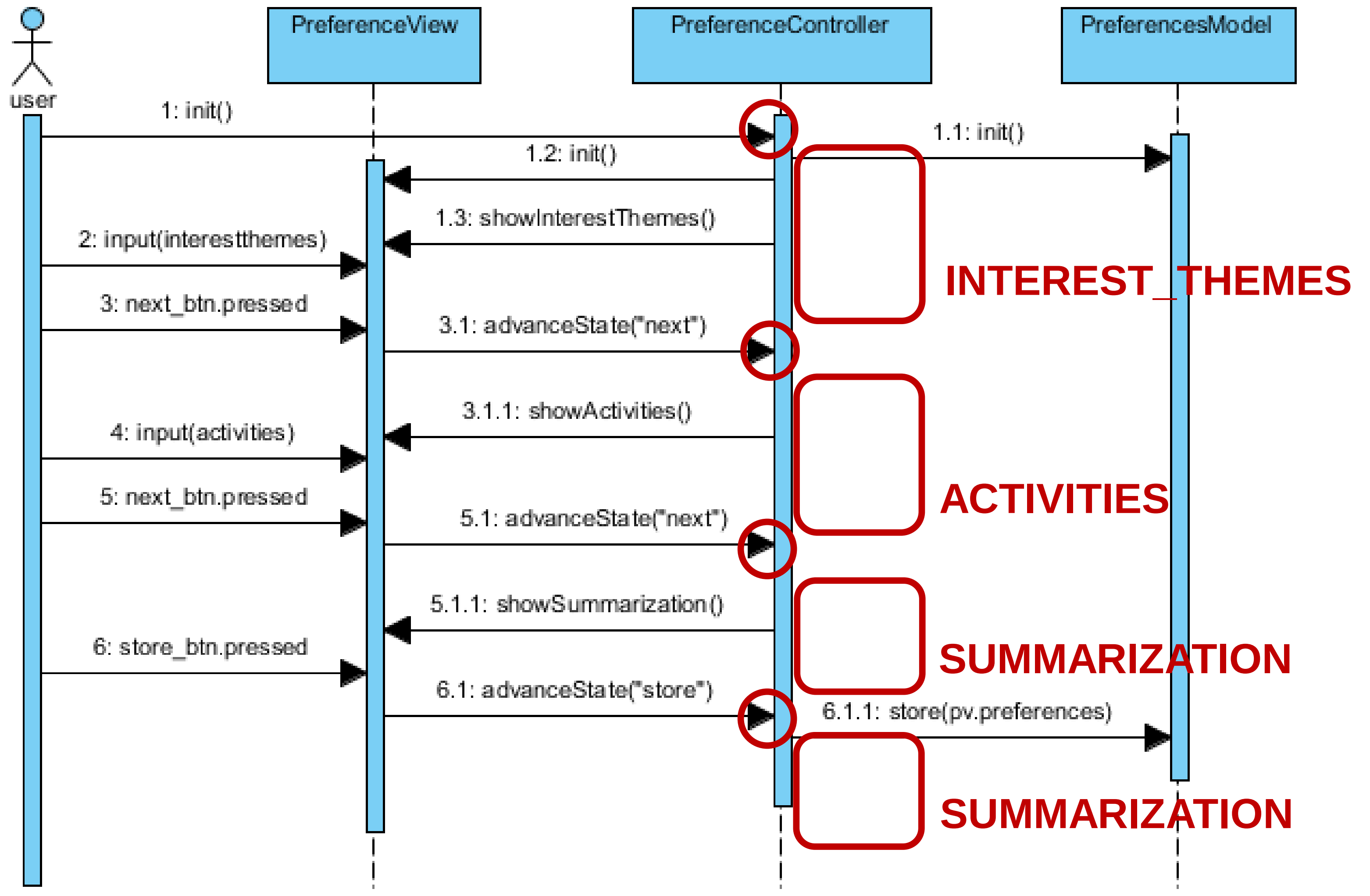
Identify states and events from scenario descriptions

Translate sequence diagrams to state charts

Integrate state charts

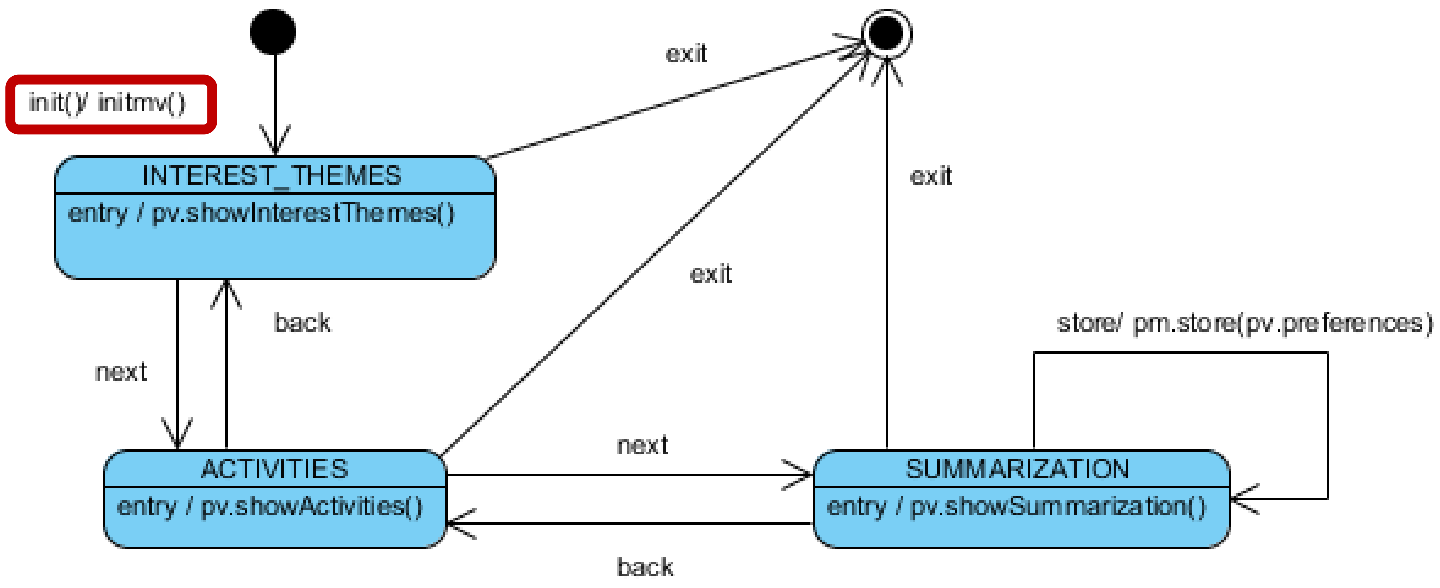
Preferences SD

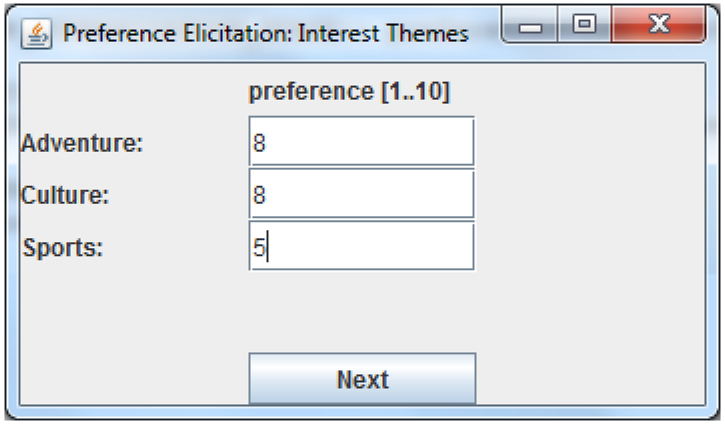
Graz University of Technology



Preferences State Chart

Graz University of Technology


init()/ initmv()
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
back
next
exit
exit
store/ pm.store(pv.preferences)
ACTIVITIES
entry / pv.showActivities()
next
SUMMARIZATION
entry / pv.showSummarization()
back

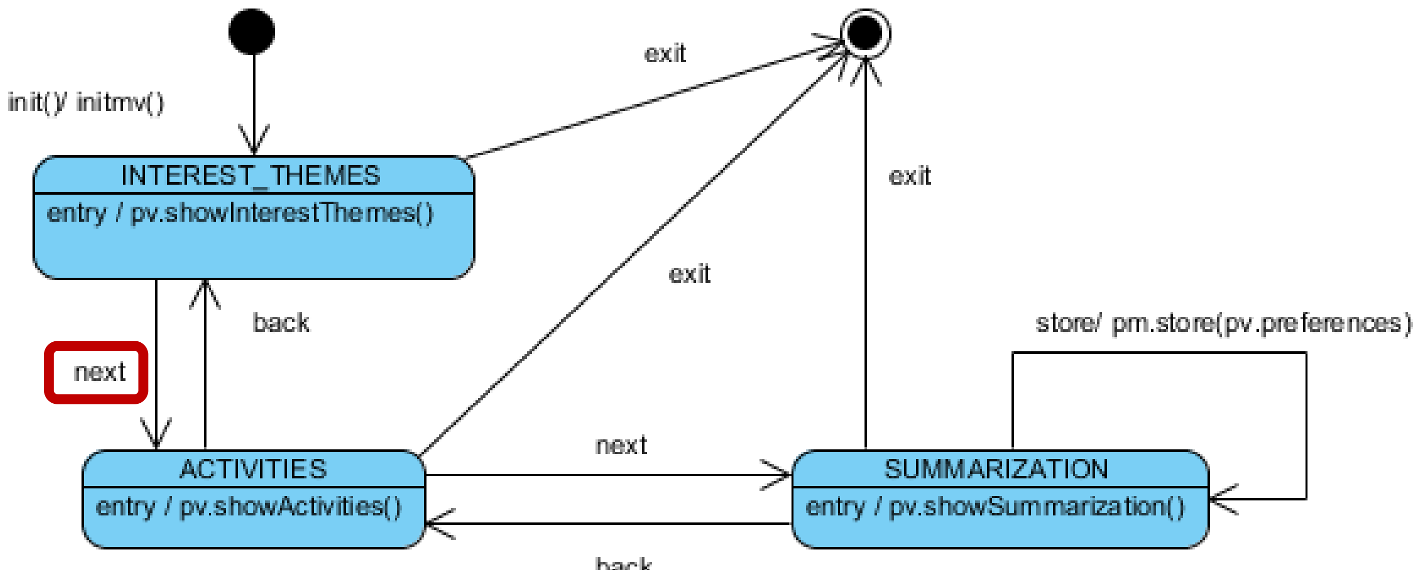


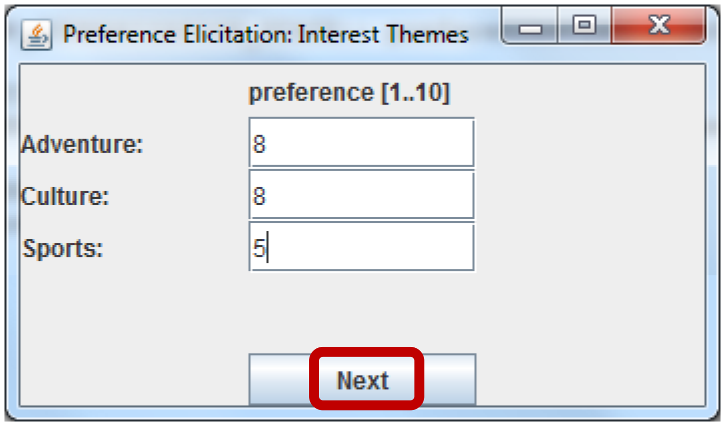
preference [1..10]

Adventure:	8
Culture:	8
Sports:	5

Preferences State Chart

Graz University of Technology


init()/ initmv()
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
next
back
exit
exit
ACTIVITIES
entry / pv.showActivities()
next
SUMMARIZATION
entry / pv.showSummarization()
store/ pm.store(pv.preferences)
back

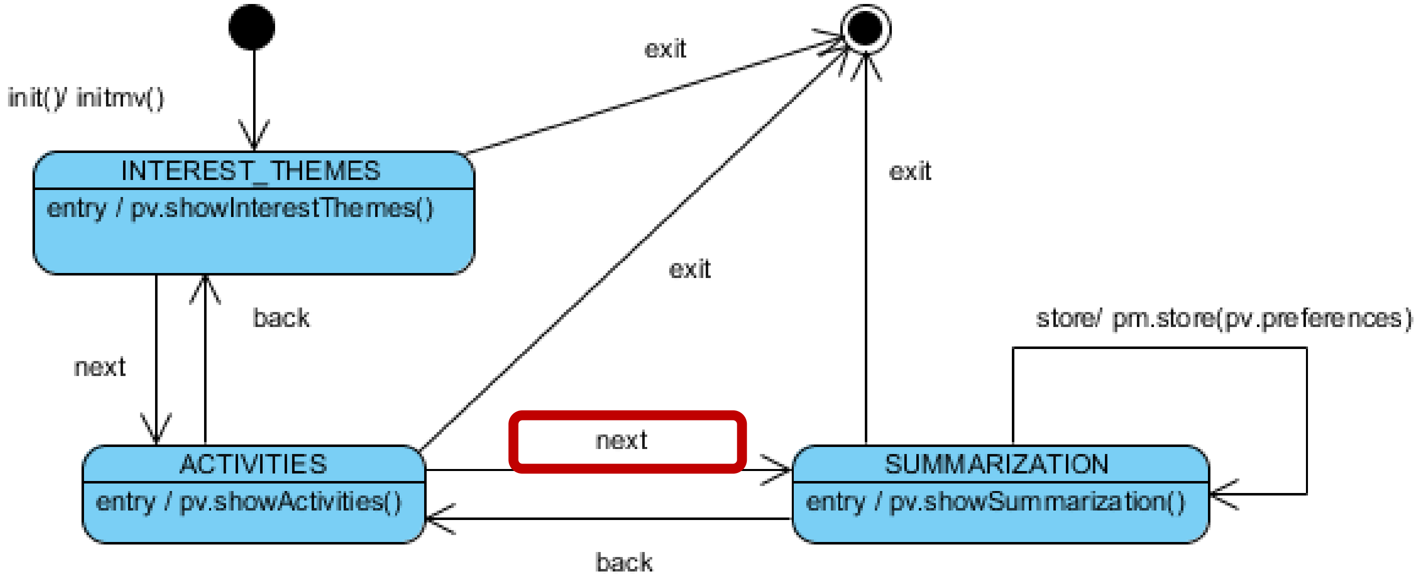


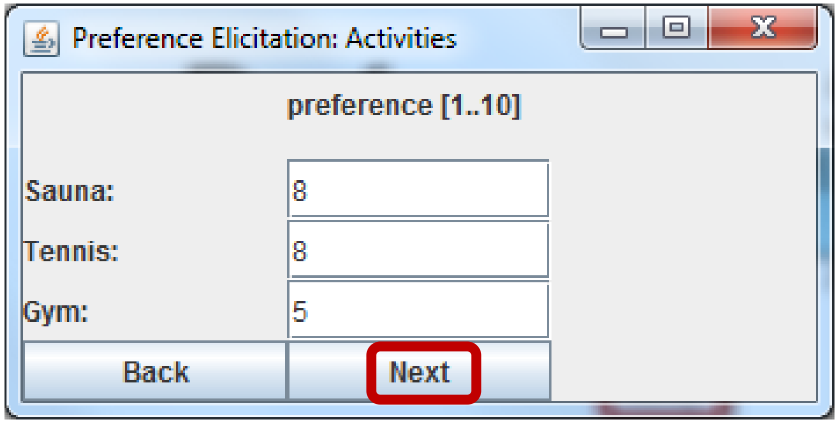
preference [1..10]

Adventure:	8
Culture:	8
Sports:	5

Preferences State Chart

Graz University of Technology


init()/ initmv()
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
next
back
exit
exit
ACTIVITIES
entry / pv.showActivities()
next
SUMMARIZATION
entry / pv.showSummarization()
store/ pm.store(pv.preferences)
back

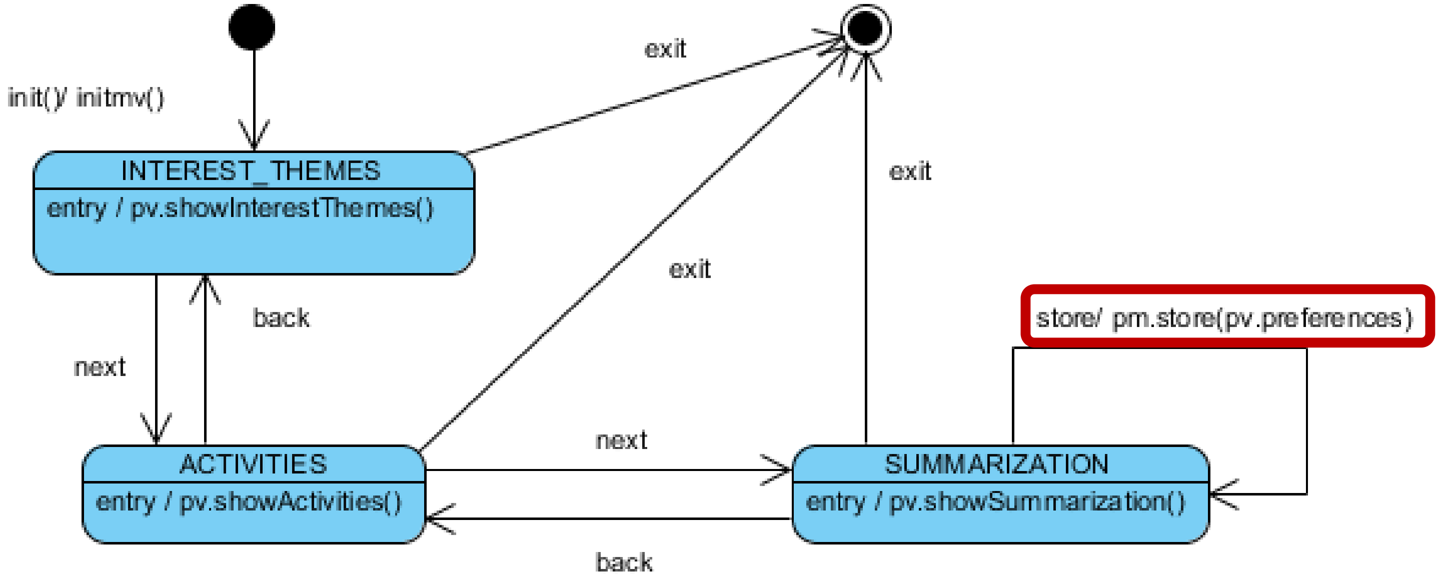

Preference Elicitation: Activities

preference [1..10]

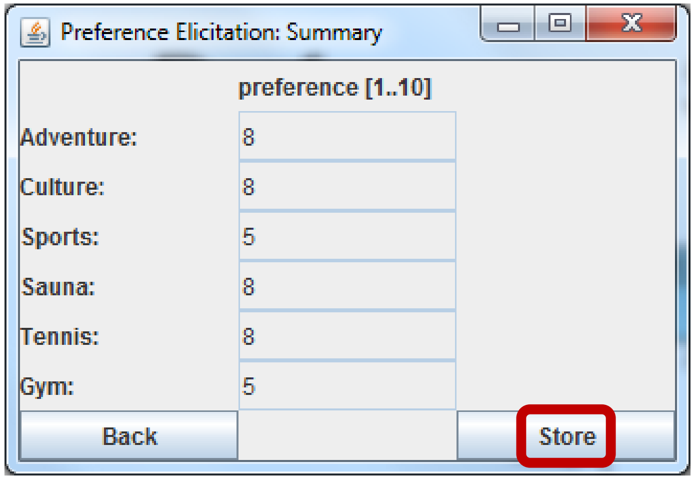
Sauna:	8
Tennis:	8
Gym:	5

Preferences State Chart

Graz University of Technology



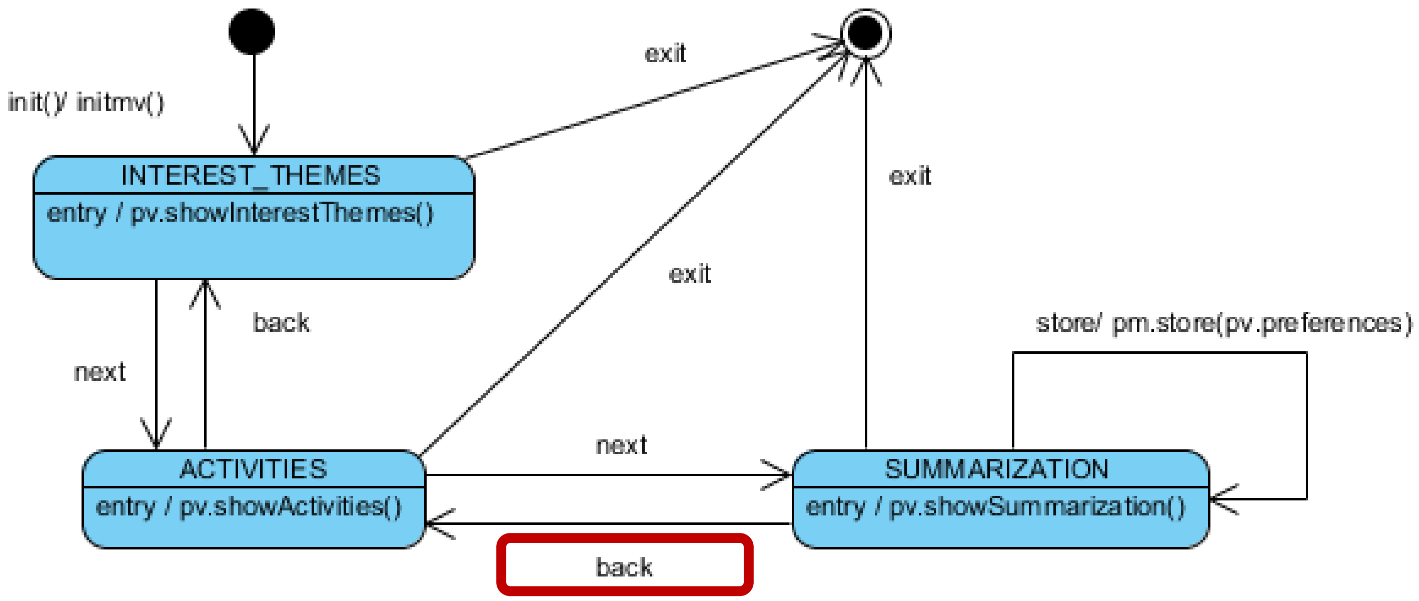
init()/ initmv()
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
exit
exit
next
back
ACTIVITIES
entry / pv.showActivities()
next
back
store/ pm.store(pv.preferences)
SUMMARIZATION
entry / pv.showSummarization()

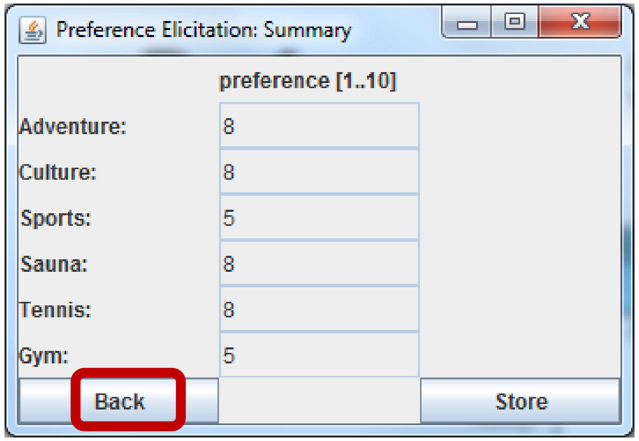


	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Preferences State Chart

Graz University of Technology

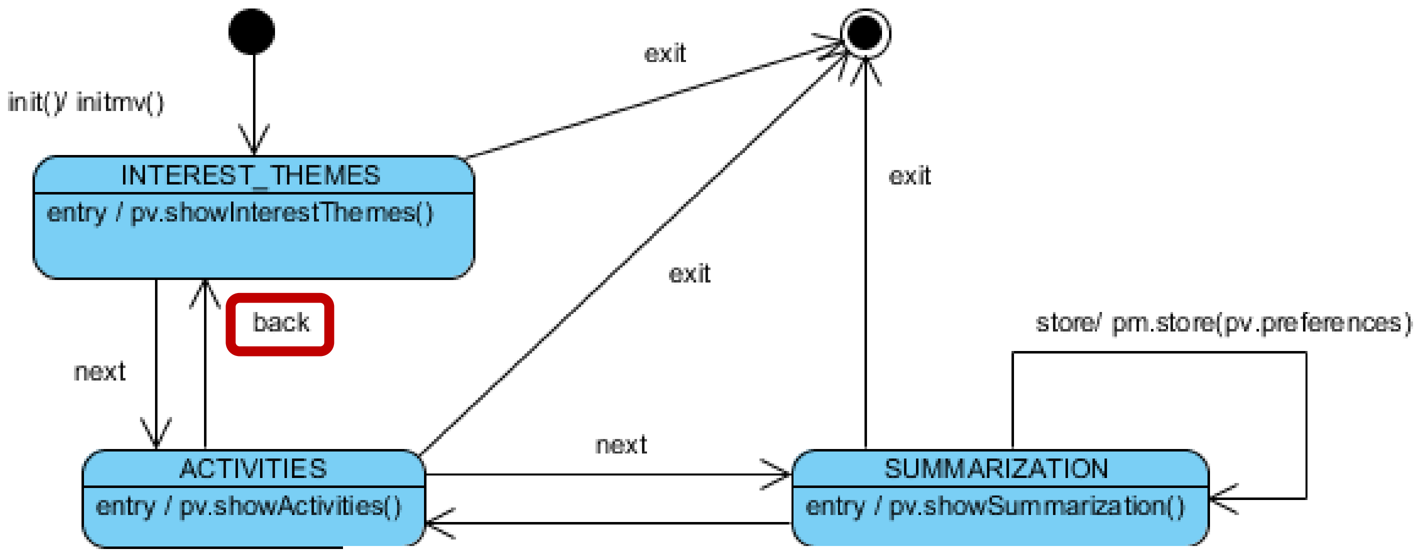

init()/ initmv()
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
exit
exit
back
next
store/ pm.store(pv.preferences)
ACTIVITIES
entry / pv.showActivities()
next
SUMMARIZATION
entry / pv.showSummarization()
back

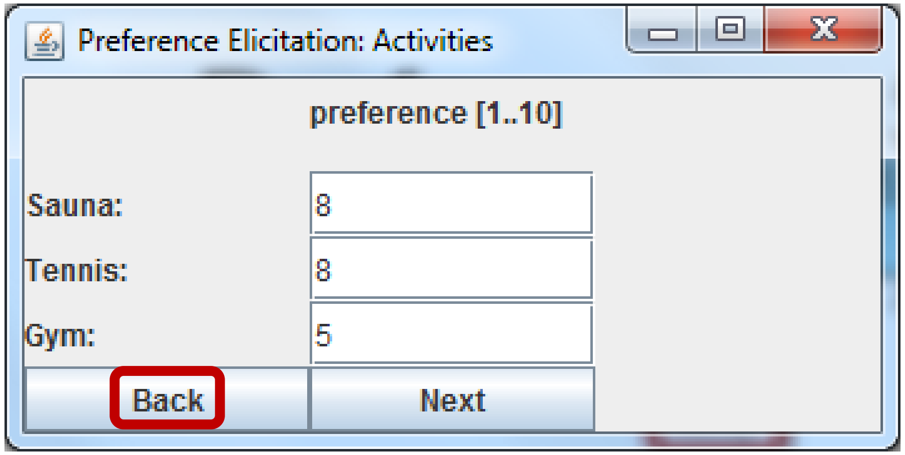


	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Preferences State Chart

Graz University of Technology

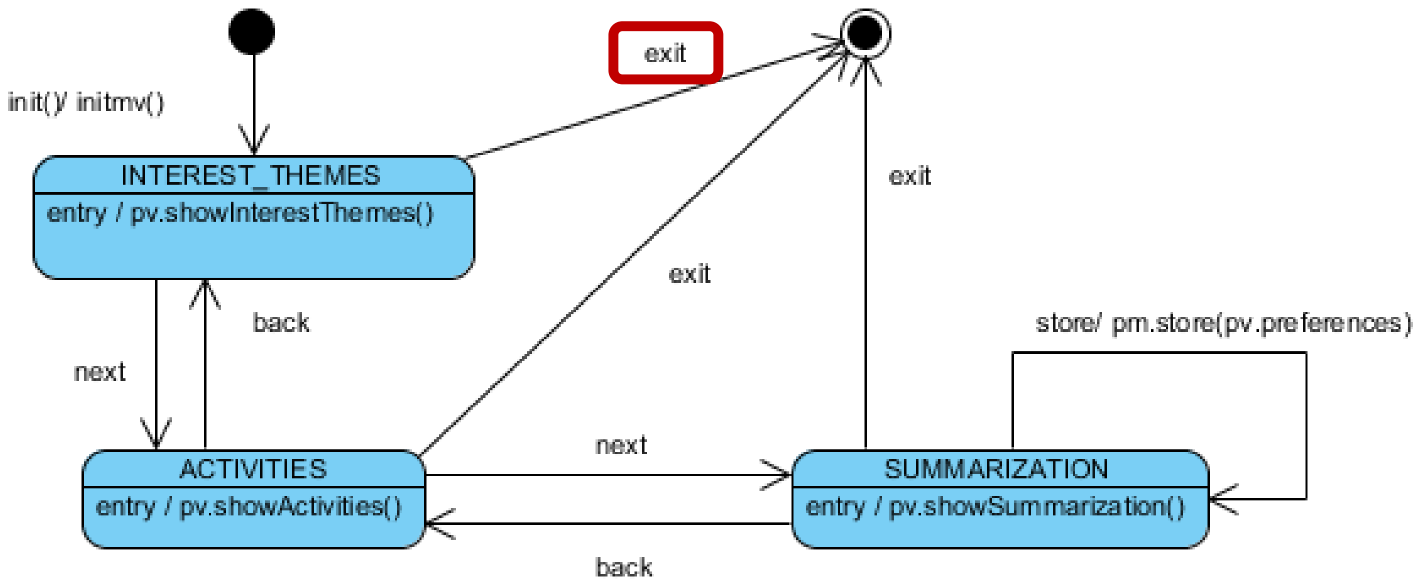

init()/ initmv()
exit
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
exit
back
next
store/ pm.store(pv.preferences)
next
ACTIVITIES
entry / pv.showActivities()
SUMMARIZATION
entry / pv.showSummarization()



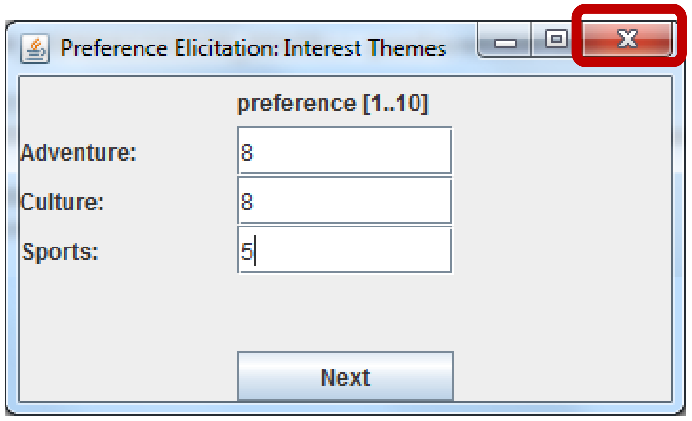
preference [1..10]	
Sauna:	8
Tennis:	8
Gym:	5

Preferences State Chart

Graz University of Technology



init()/ initmv()
exit
INTEREST_THEMES
entry / pv.showInterestThemes()
exit
exit
back
next
store/ pm.store(pv.preferences)
next
ACTIVITIES
entry / pv.showActivities()
SUMMARIZATION
entry / pv.showSummarization()
back

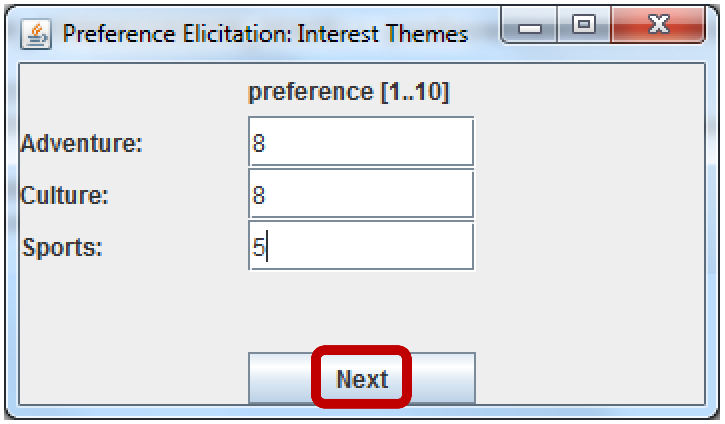

Preference Elicitation: Interest Themes

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5

Example Java Code (Alternative 1)

Graz University of Technology

```
public void advanceState(String event){Integer s = getState();
      switch (event) {
         case "next":  if (s == INTEREST_THEMES_STATE)
               {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                              else {setState(SHOW_SUMMARIZATION_STATE);
                                      preferencesview.showSummarization();} break;
         case "back":  if (s == ACTIVITIES_STATE)
               {setState(INTEREST_THEMES_STATE);
                              preferencesview.showInterestThemes();}
               else {setState(ACTIVITIES_STATE);
                              preferencesview.showActivities();} break;
         case "store":  if (s == SHOW_SUMMARIZATION_STATE)
               {preferencesmodel.store();} break;
         default: break;
      }
   }
```

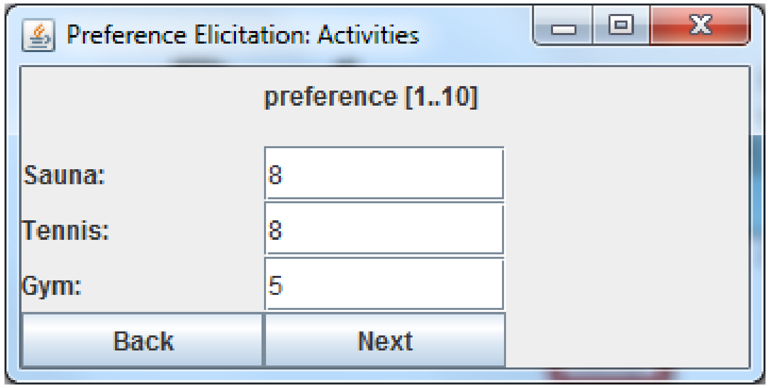

Preference Elicitation: Interest Themes

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5

Example Java Code (Alternative 1)

Graz University of Technology

```
public void advanceState(String event){Integer s = getState();
    switch (event) {
      case "next":  if (s == INTEREST_THEMES_STATE)
          {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                  else {setState(SHOW_SUMMARIZATION_STATE);
                        preferencesview.showSummarization();} break;
      case "back":  if (s == ACTIVITIES_STATE)
          {setState(INTEREST_THEMES_STATE);
                  preferencesview.showInterestThemes();}
          else {setState(ACTIVITIES_STATE);
                        preferencesview.showActivities();} break;
      case "store":  if (s == SHOW_SUMMARIZATION_STATE)
          {preferencesmodel.store();} break;
      default: break;
    }
  }
```



Preference Elicitation: Activities

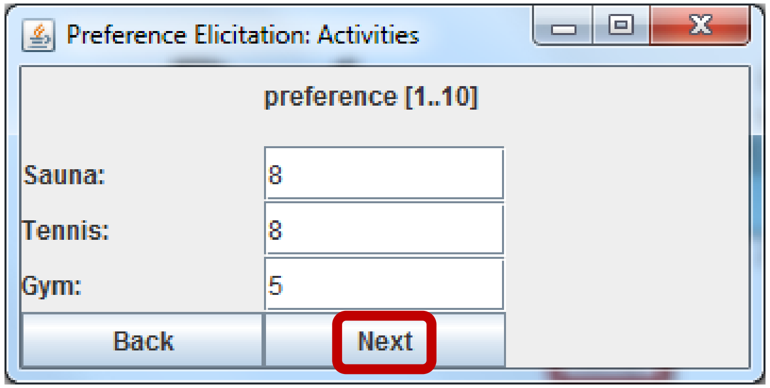
preference [1..10]

Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

Graz University of Technology

```
public void advanceState(String event){Integer s = getState();
      switch (event) {
        case "next":  if (s == INTEREST_THEMES_STATE)
              {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                              else {setState(SHOW_SUMMARIZATION_STATE);
                                      preferencesview.showSummarization();} break;
        case "back":  if (s == ACTIVITIES_STATE)
              {setState(INTEREST_THEMES_STATE);
                              preferencesview.showInterestThemes();}
              else {setState(ACTIVITIES_STATE);
                                      preferencesview.showActivities();} break;
        case "store":  if (s == SHOW_SUMMARIZATION_STATE)
              {preferencesmodel.store();} break;
        default: break;
      }
   }
```

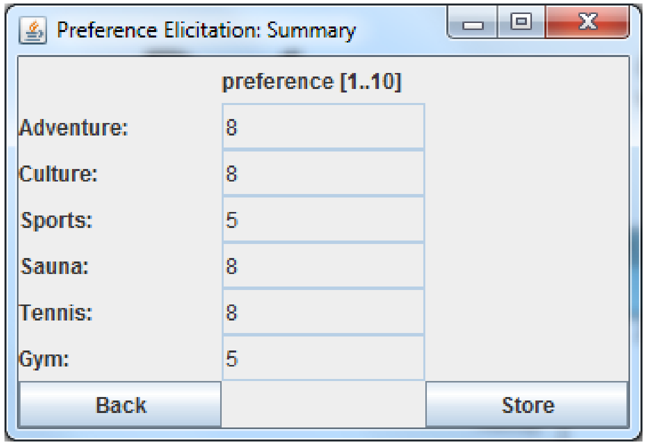


preference [1..10]

Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

```
public void advanceState(String event){Integer s = getState();
    switch (event) {
      case "next":  if (s == INTEREST_THEMES_STATE)
          {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                    else {setState(SHOW_SUMMARIZATION_STATE);
                          preferencesview.showSummarization();} break;
      case "back":  if (s == ACTIVITIES_STATE)
          {setState(INTEREST_THEMES_STATE);
                    preferencesview.showInterestThemes();}
          else {setState(ACTIVITIES_STATE);
                          preferencesview.showActivities();} break;
      case "store":  if (s == SHOW_SUMMARIZATION_STATE)
          {preferencesmodel.store();} break;
      default: break;
    }
  }
```

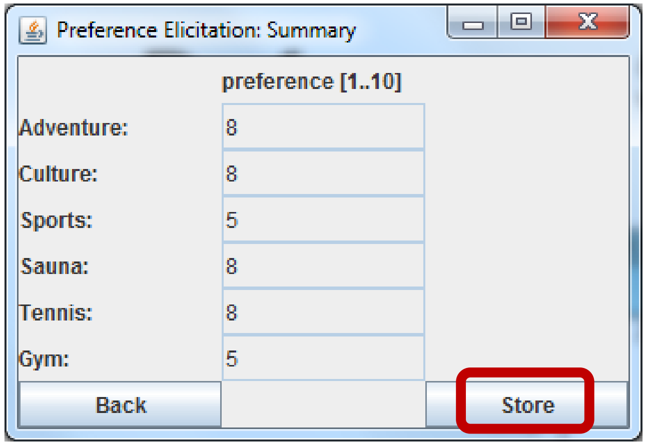


Preference Elicitation: Summary

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

```
public void advanceState(String event){Integer s = getState();
      switch (event) {
         case "next":  if (s == INTEREST_THEMES_STATE)
             {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                          else {setState(SHOW_SUMMARIZATION_STATE);
                                 preferencesview.showSummarization();} break;
         case "back":  if (s == ACTIVITIES_STATE)
             {setState(INTEREST_THEMES_STATE);
                          preferencesview.showInterestThemes();}
             else {setState(ACTIVITIES_STATE);
                                 preferencesview.showActivities();} break;
         case "store":  if (s == SHOW_SUMMARIZATION_STATE)
             {preferencesmodel.store();} break;
         default: break;
      }
   }
```

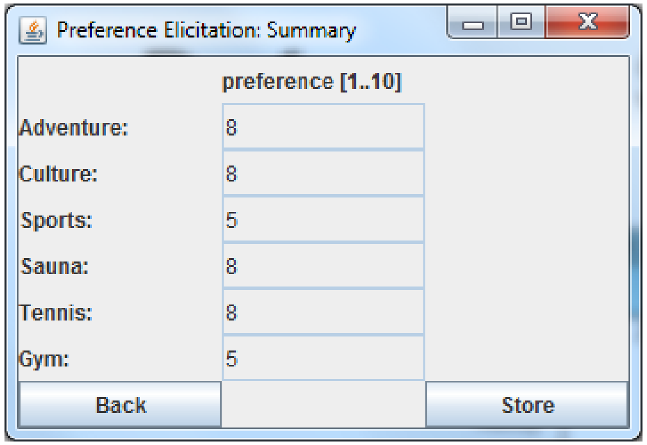

Preference Elicitation: Summary

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

Graz University of Technology

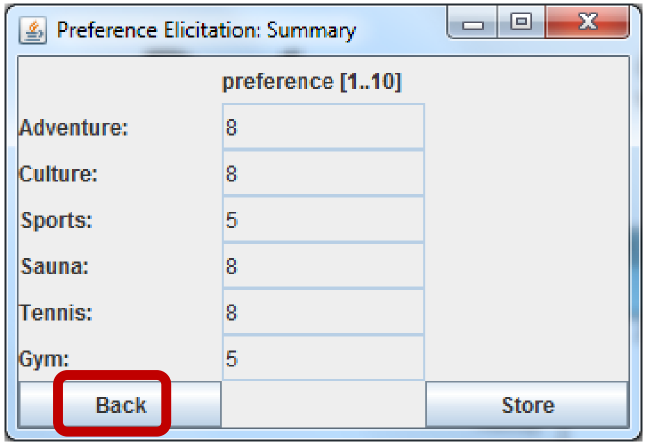
```
public void advanceState(String event){Integer s = getState();
      switch (event) {
         case "next":  if (s == INTEREST_THEMES_STATE)
               {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                              else {setState(SHOW_SUMMARIZATION_STATE);
                                     preferencesview.showSummarization();} break;
         case "back":  if (s == ACTIVITIES_STATE)
               {setState(INTEREST_THEMES_STATE);
                              preferencesview.showInterestThemes();}
               else {setState(ACTIVITIES_STATE);
                                     preferencesview.showActivities();} break;
         case "store":  if (s == SHOW_SUMMARIZATION_STATE)
               {preferencesmodel.store();} break;
         default: break;
      }
   }
```



	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

```
public void advanceState(String event){Integer s = getState();
    switch (event) {
      case "next":  if (s == INTEREST_THEMES_STATE)
          {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                    else {setState(SHOW_SUMMARIZATION_STATE);
                          preferencesview.showSummarization();} break;
      case "back": if (s == ACTIVITIES_STATE)
          {setState(INTEREST_THEMES_STATE);
                    preferencesview.showInterestThemes();}
          else {setState(ACTIVITIES_STATE);
                          preferencesview.showActivities();} break;
      case "store":  if (s == SHOW_SUMMARIZATION_STATE)
          {preferencesmodel.store();} break;
      default: break;
    }
  }
```

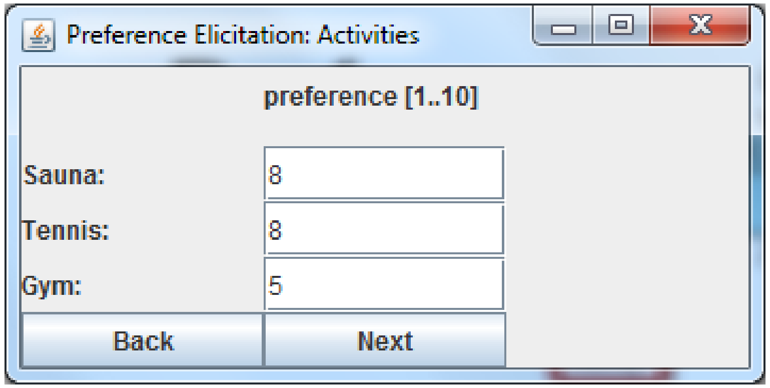

Preference Elicitation: Summary

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5
Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

Graz University of Technology

```
public void advanceState(String event){Integer s = getState();
    switch (event) {
      case "next":  if (s == INTEREST_THEMES_STATE)
          {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                    else {setState(SHOW_SUMMARIZATION_STATE);
                          preferencesview.showSummarization();} break;
      case "back":  if (s == ACTIVITIES_STATE)
          {setState(INTEREST_THEMES_STATE);
                    preferencesview.showInterestThemes();}
          else {setState(ACTIVITIES_STATE);
                          preferencesview.showActivities();} break;
      case "store":  if (s == SHOW_SUMMARIZATION_STATE)
          {preferencesmodel.store();} break;
      default: break;
    }
  }
```

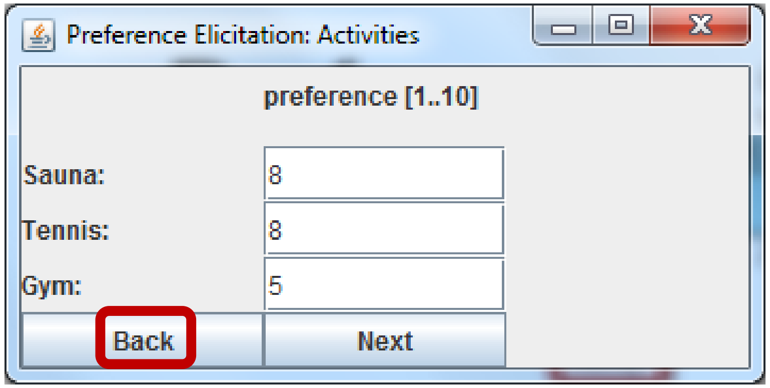

Preference Elicitation: Activities

preference [1..10]

Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

```
public void advanceState(String event){Integer s = getState();
      switch (event) {
         case "next":  if (s == INTEREST_THEMES_STATE)
               {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                           else {setState(SHOW_SUMMARIZATION_STATE);
                                  preferencesview.showSummarization();} break;
         case "back":  if (s == ACTIVITIES_STATE)
               {setState(INTEREST_THEMES_STATE);
                           preferencesview.showInterestThemes();}
               else {setState(ACTIVITIES_STATE);
                                  preferencesview.showActivities();} break;
         case "store":  if (s == SHOW_SUMMARIZATION_STATE)
               {preferencesmodel.store();} break;
         default: break;
      }
   }
```


Preference Elicitation: Activities

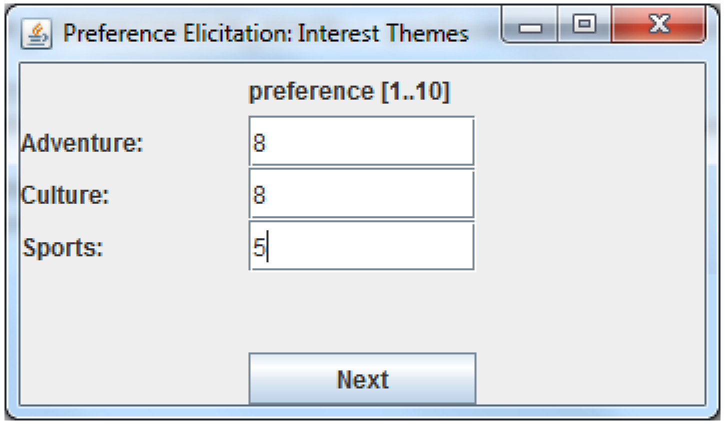
preference [1..10]

Sauna:	8
Tennis:	8
Gym:	5

Example Java Code (Alternative 1)

Graz University of Technology

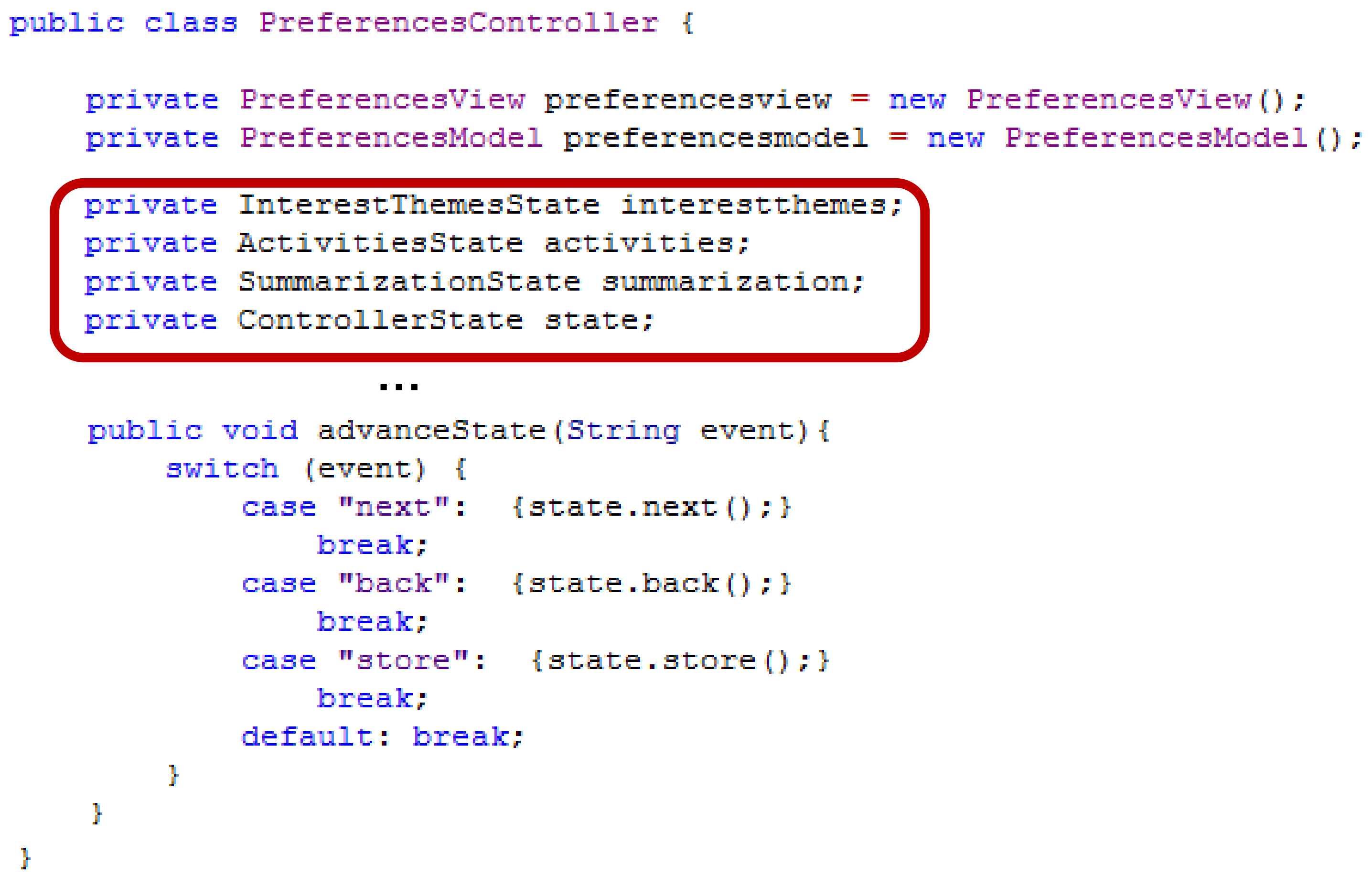
```
public void advanceState(String event){Integer s = getState();
    switch (event) {
      case "next":  if (s == INTEREST_THEMES_STATE)
          {setState(ACTIVITIES_STATE); preferencesview.showActivities();}
                    else {setState(SHOW_SUMMARIZATION_STATE);
                          preferencesview.showSummarization();} break;
      case "back":  if (s == ACTIVITIES_STATE)
          {setState(INTEREST_THEMES_STATE);
                    preferencesview.showInterestThemes();}
          else {setState(ACTIVITIES_STATE);
                          preferencesview.showActivities();} break;
      case "store":  if (s == SHOW_SUMMARIZATION_STATE)
          {preferencesmodel.store();} break;
      default: break;
    }
  }
```


Preference Elicitation: Interest Themes

	preference [1..10]
Adventure:	8
Culture:	8
Sports:	5

Example Java Code (Alternative 2)

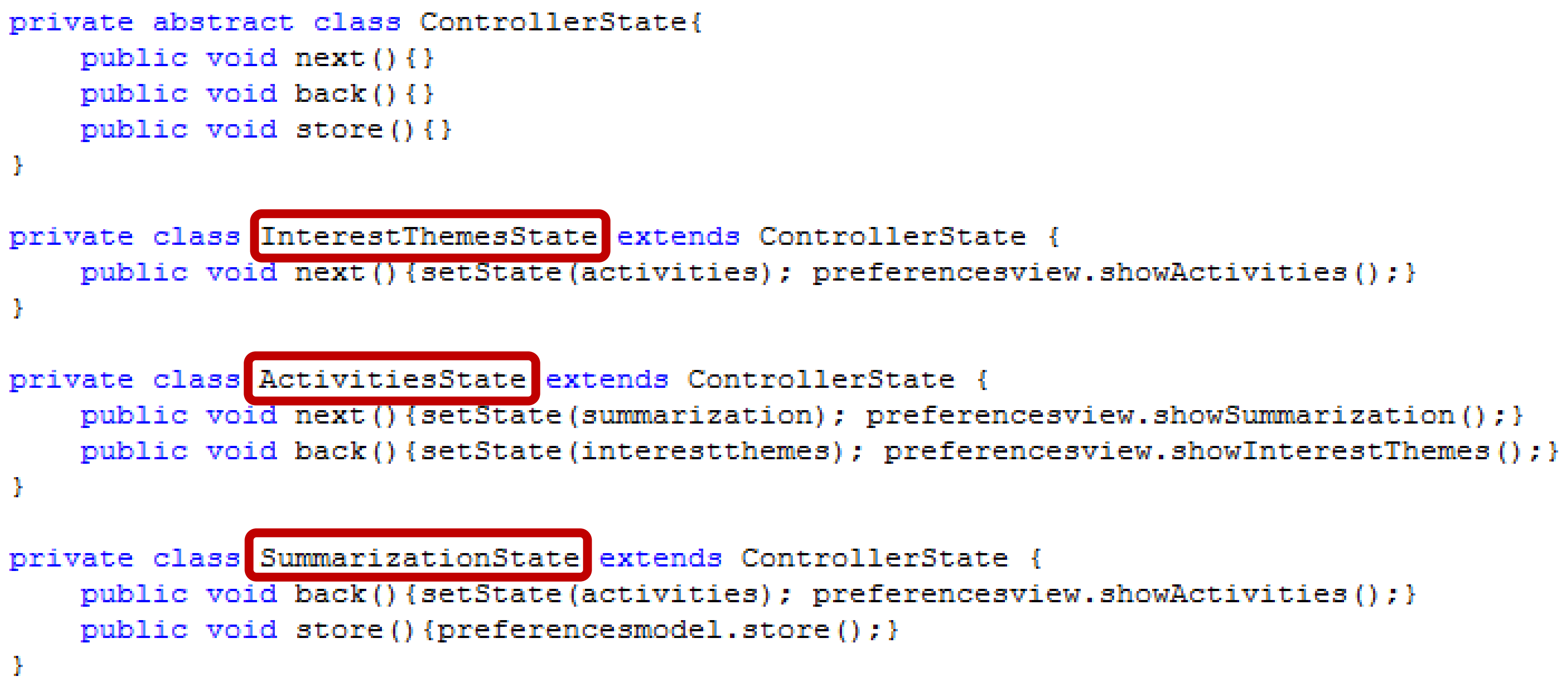
Graz University of Technology



```
public class PreferencesController {

    private PreferencesView preferencesview = new PreferencesView();
    private PreferencesModel preferencesmodel = new PreferencesModel();

    private InterestThemesState interestthemes;
    private ActivitiesState activities;
    private SummarizationState summarization;
    private ControllerState state;
                    ...
    public void advanceState(String event){
        switch (event) {
            case "next":  {state.next();}
                break;
            case "back":  {state.back();}
                break;
            case "store":  {state.store();}
                break;
            default: break;
        }
    }
}
```

Example Java Code (Alternative 2)

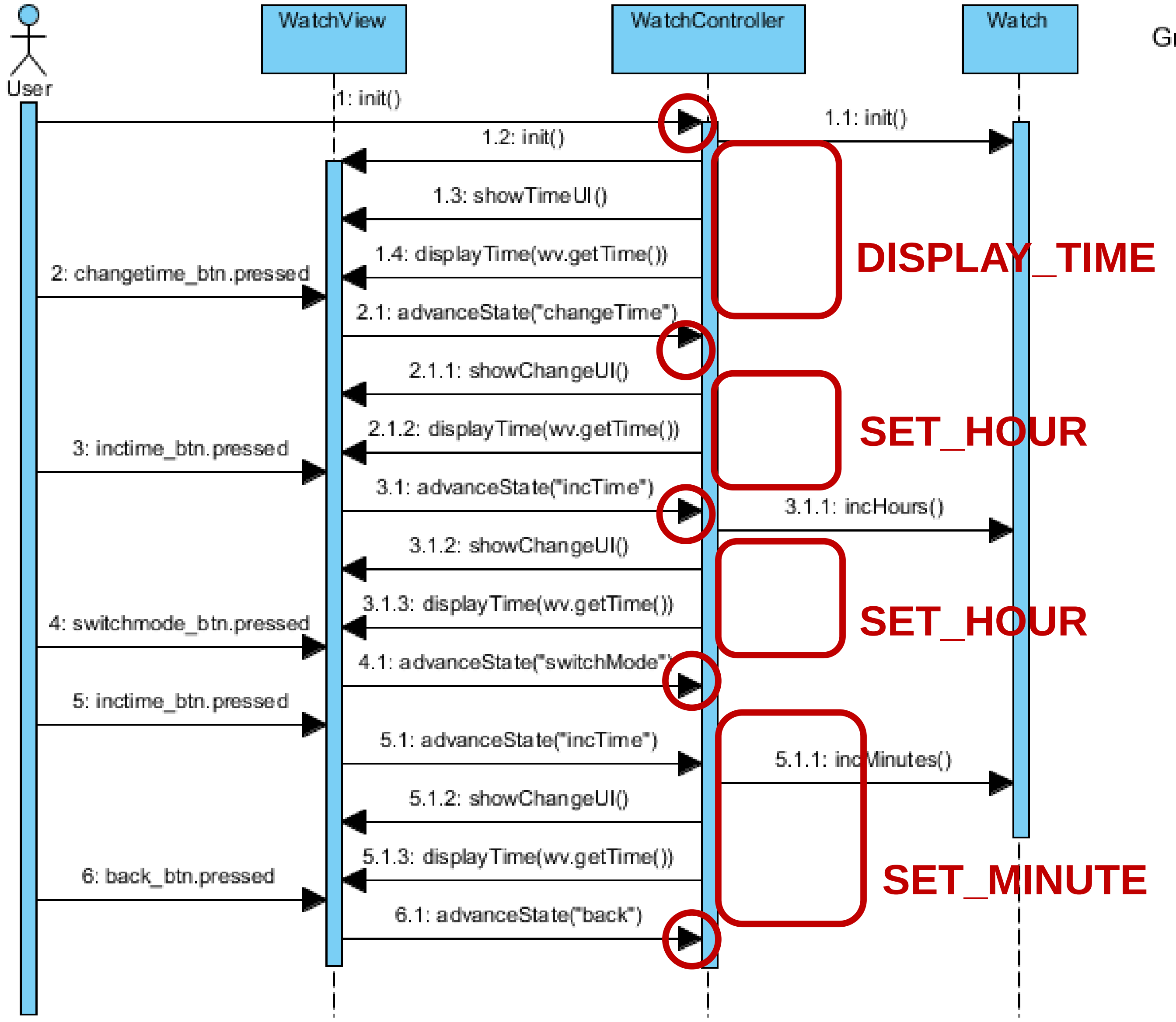
TU Graz
Graz University of Technology



```java
private abstract class ControllerState{
    public void next(){}
    public void back(){}
    public void store(){}
}

private class InterestThemesState extends ControllerState {
    public void next(){setState(activities); preferencesview.showActivities();}
}

private class ActivitiesState extends ControllerState {
    public void next(){setState(summarization); preferencesview.showSummarization();}
    public void back(){setState(interestthemes); preferencesview.showInterestThemes();}
}

private class SummarizationState extends ControllerState {
    public void back(){setState(activities); preferencesview.showActivities();}
    public void store(){preferencesmodel.store();}
}
```

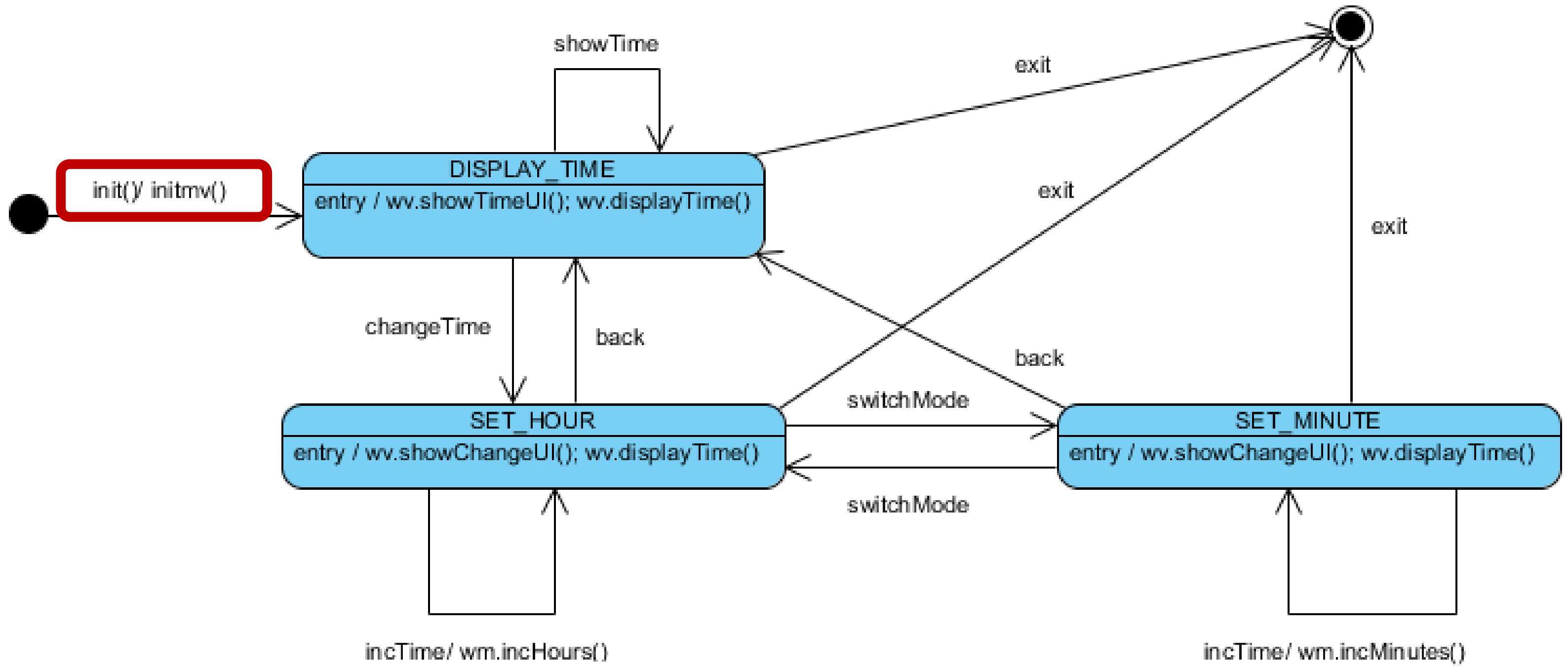
Watch SD

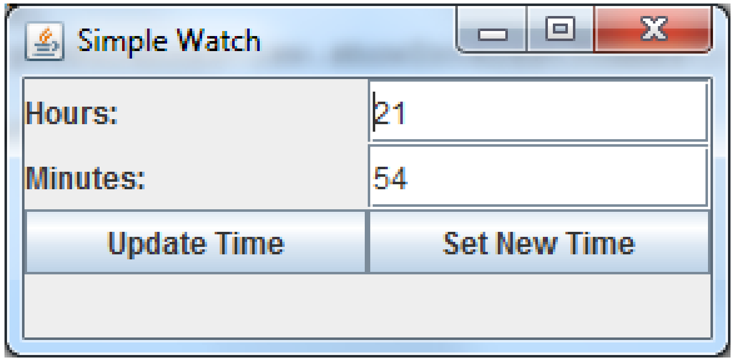
Graz University of Technology


User
WatchView
WatchController
Watch
1: init()
1.1: init()
1.2: init()
1.3: showTimeUI()
1.4: displayTime(wv.getTime())
2: changetime_btn.pressed
2.1: advanceState("changeTime")
2.1.1: showChangeUI()
2.1.2: displayTime(wv.getTime())
3: inctime_btn.pressed
3.1: advanceState("incTime")
3.1.1: incHours()
3.1.2: showChangeUI()
3.1.3: displayTime(wv.getTime())
4: switchmode_btn.pressed
4.1: advanceState("switchMode")
5: inctime_btn.pressed
5.1: advanceState("incTime")
5.1.1: incMinutes()
5.1.2: showChangeUI()
5.1.3: displayTime(wv.getTime())
6: back_btn.pressed
6.1: advanceState("back")
DISPLAY_TIME
SET_HOUR
SET_HOUR
SET_MINUTE

Simple Watch State Chart

Graz University of Technology



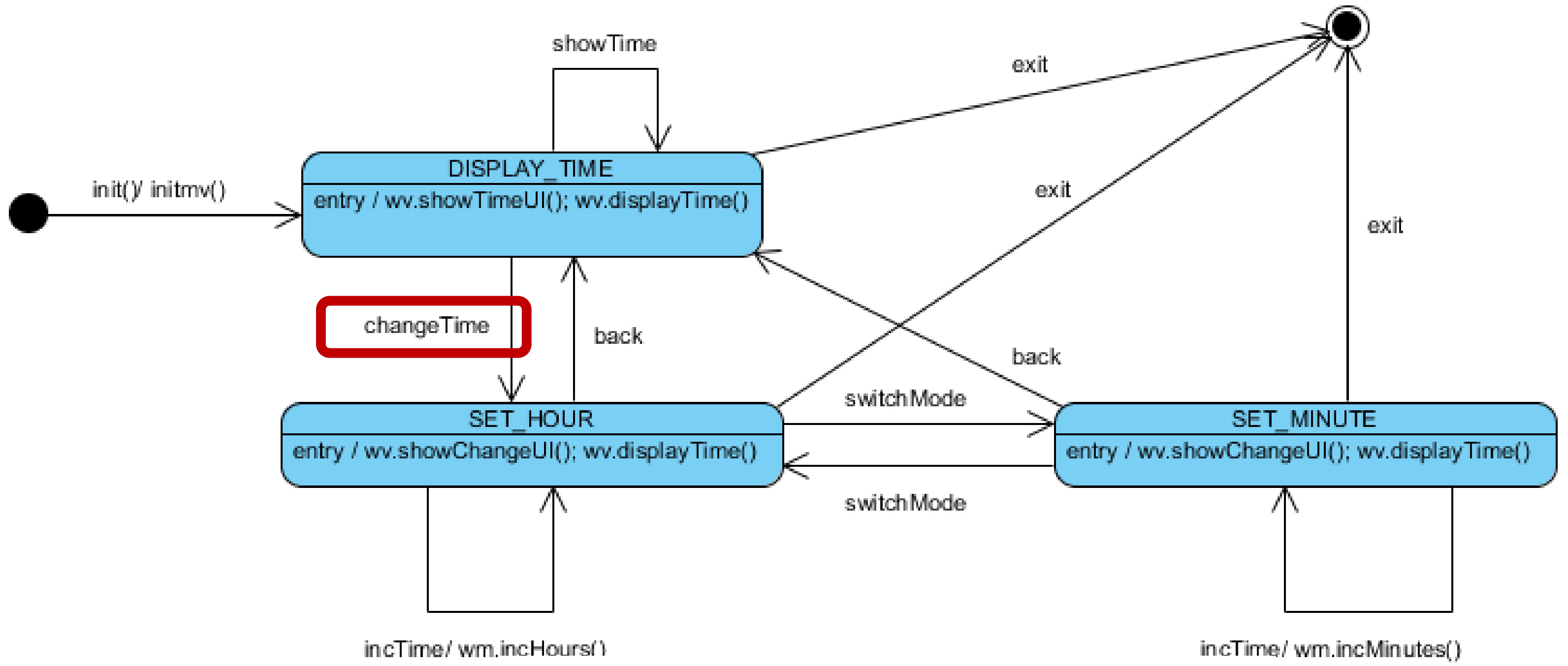


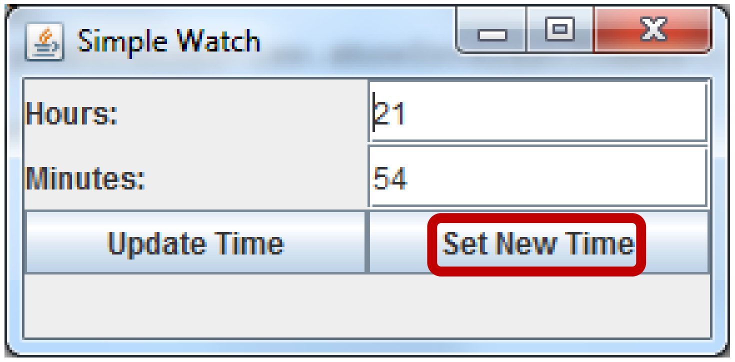
Hours: 21

Minutes: 54

Simple Watch State Chart

Graz University of Technology

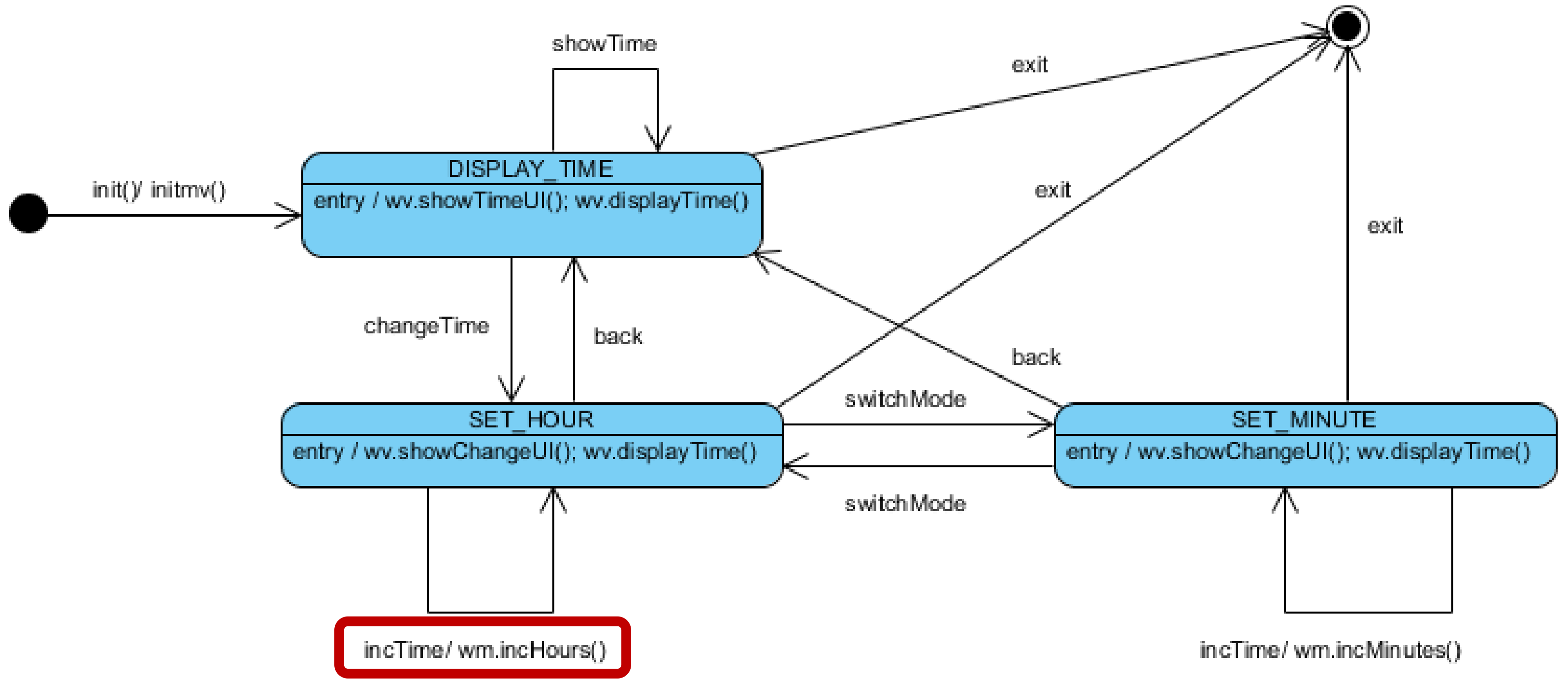


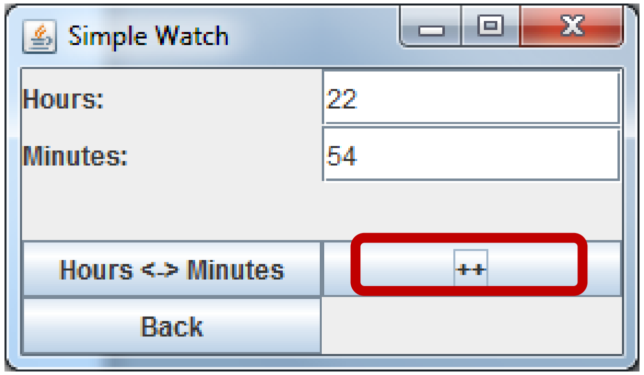


Hours: 21

Minutes: 54

Simple Watch State Chart



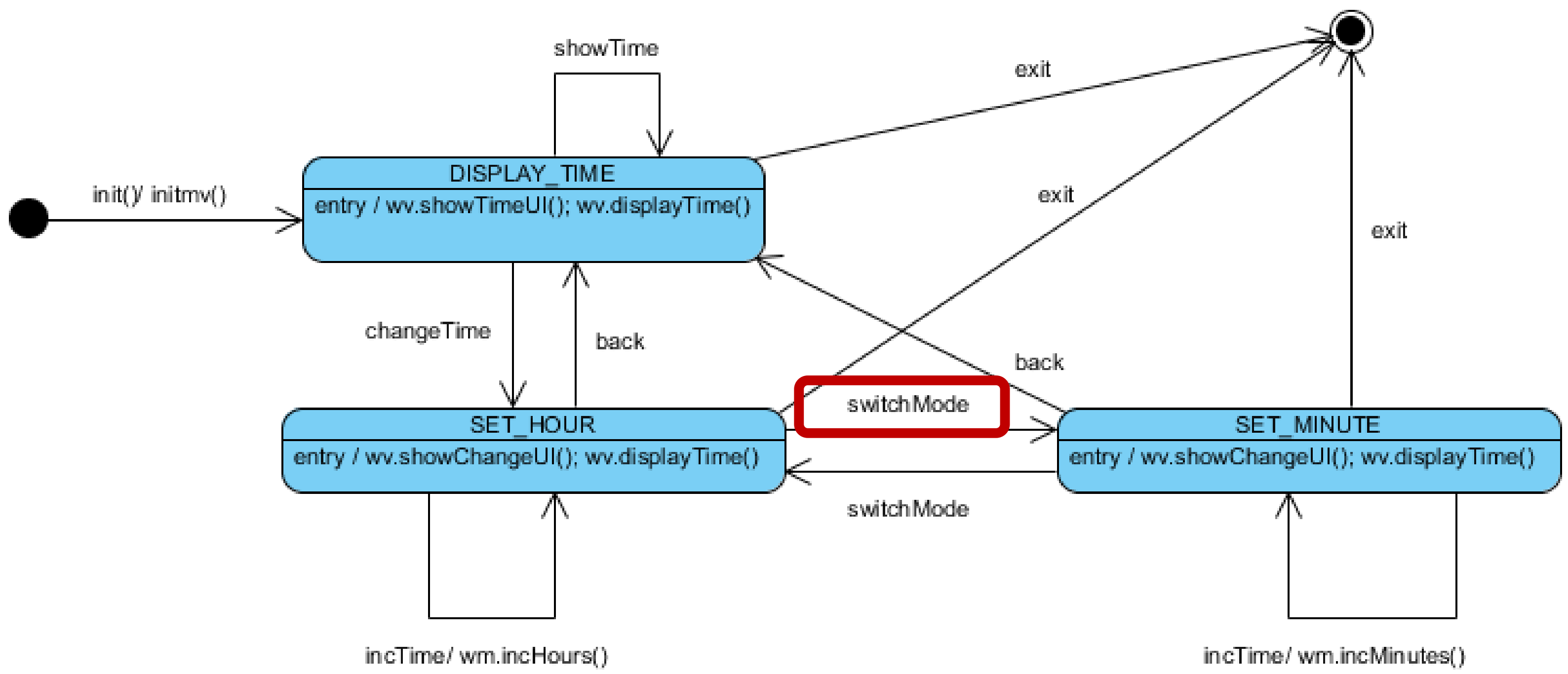


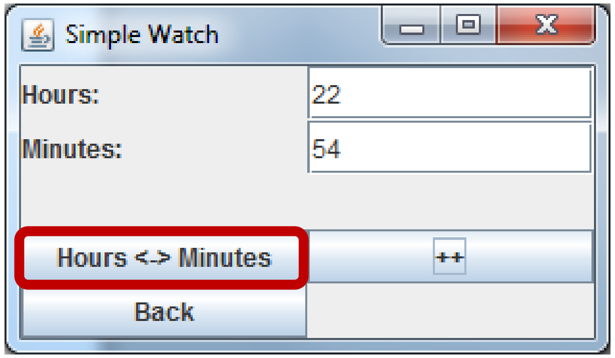
Hours: 22

Minutes: 54

Simple Watch State Chart

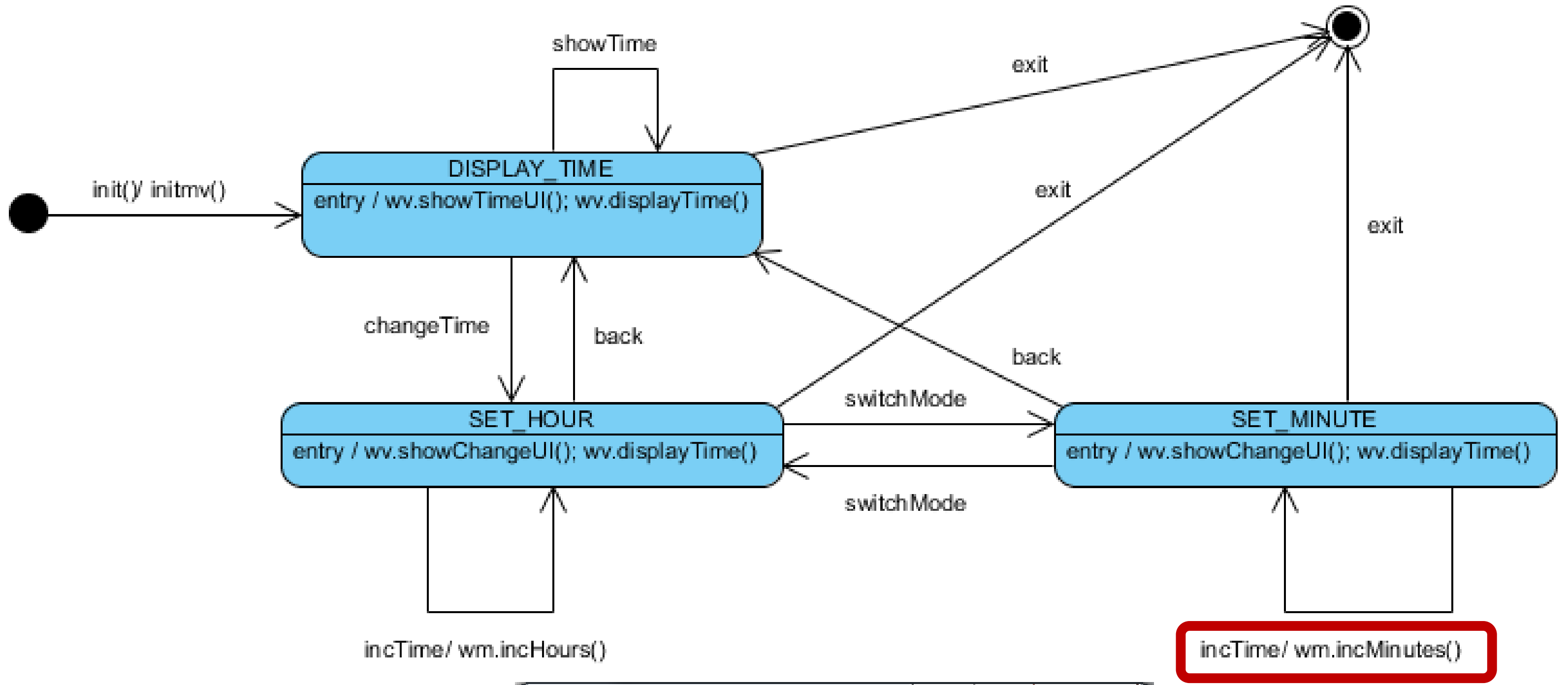
Graz University of Technology

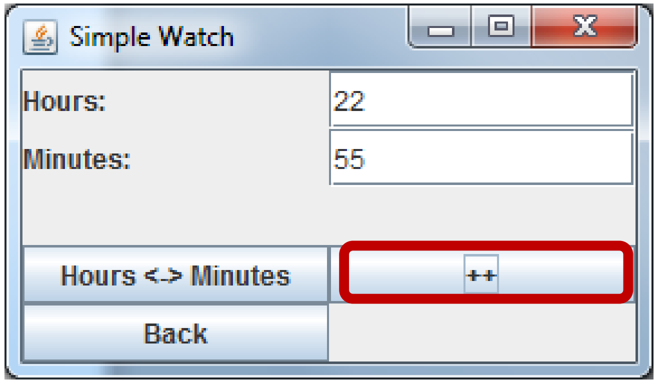




Simple Watch State Chart

Graz University of Technology



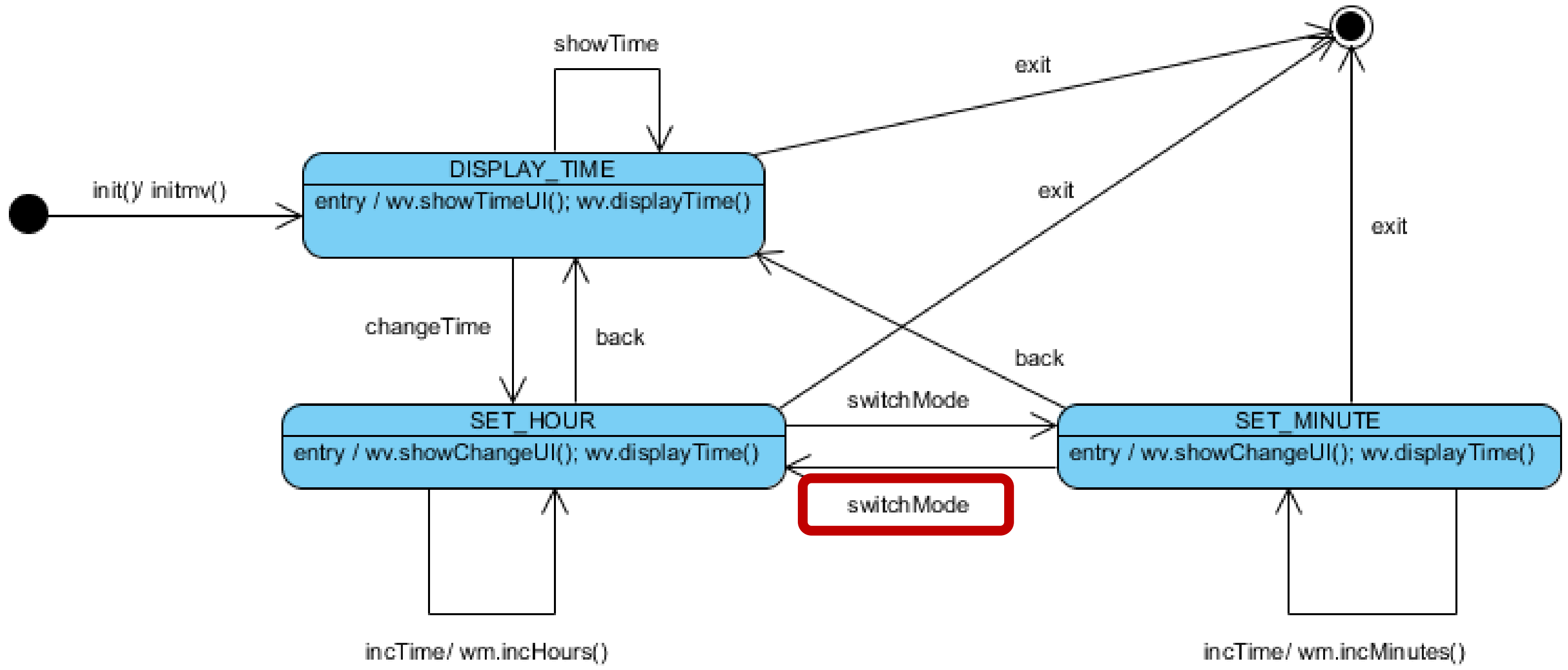


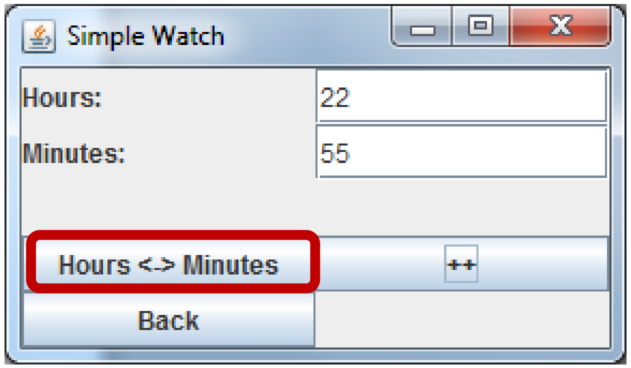
Hours: 22

Minutes: 55

Simple Watch State Chart

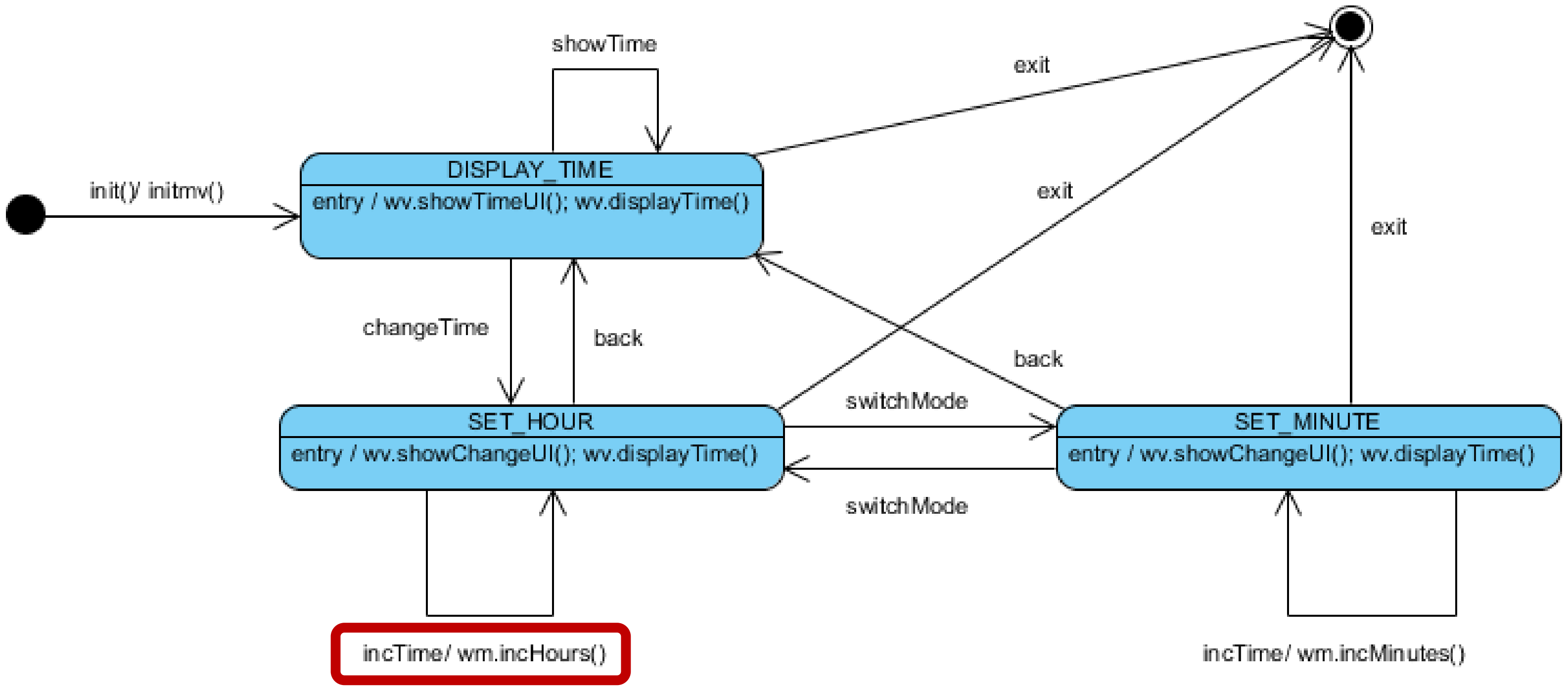
Graz University of Technology

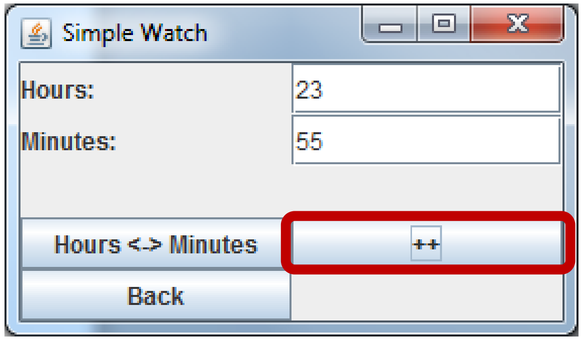




Simple Watch State Chart

Graz University of Technology



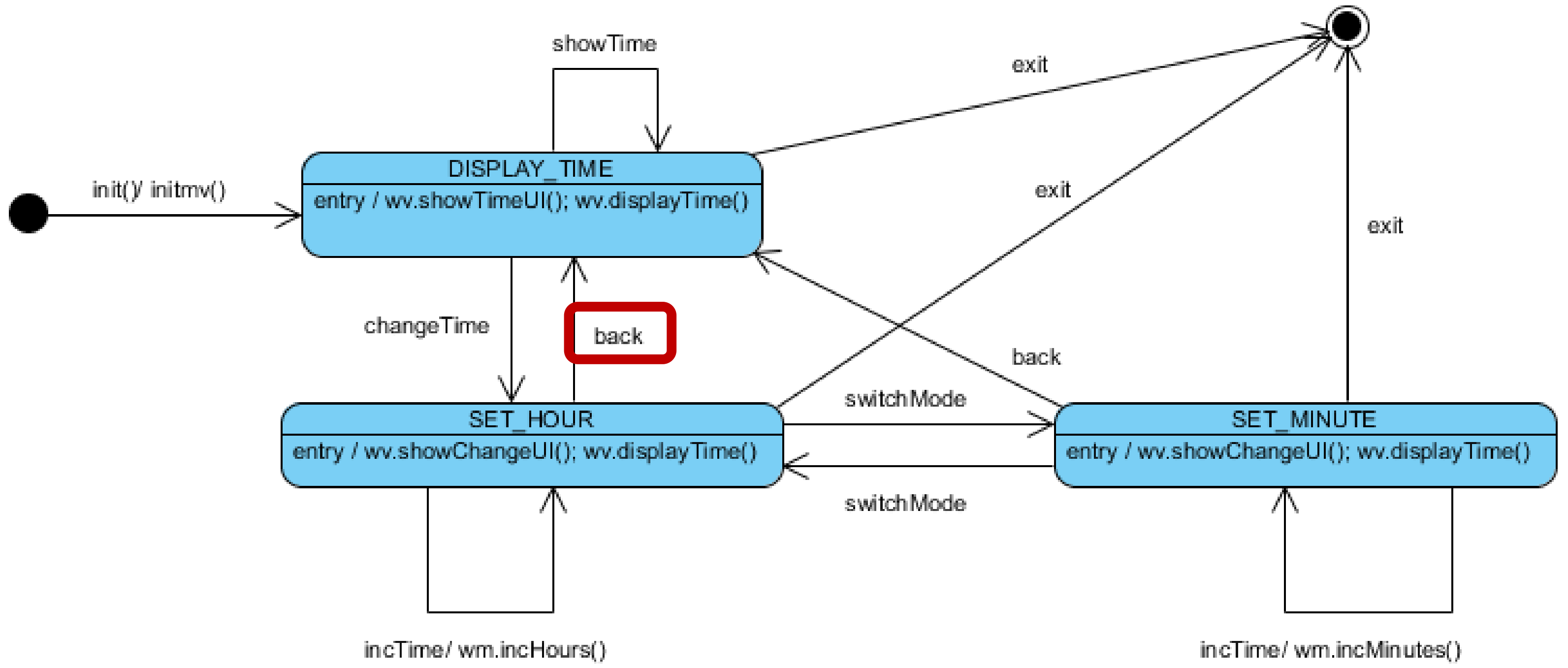


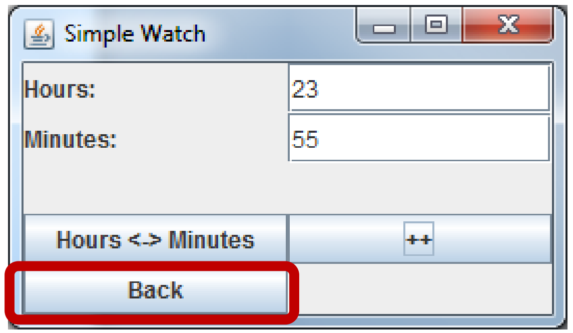
Hours: 23

Minutes: 55

Simple Watch State Chart

Graz University of Technology

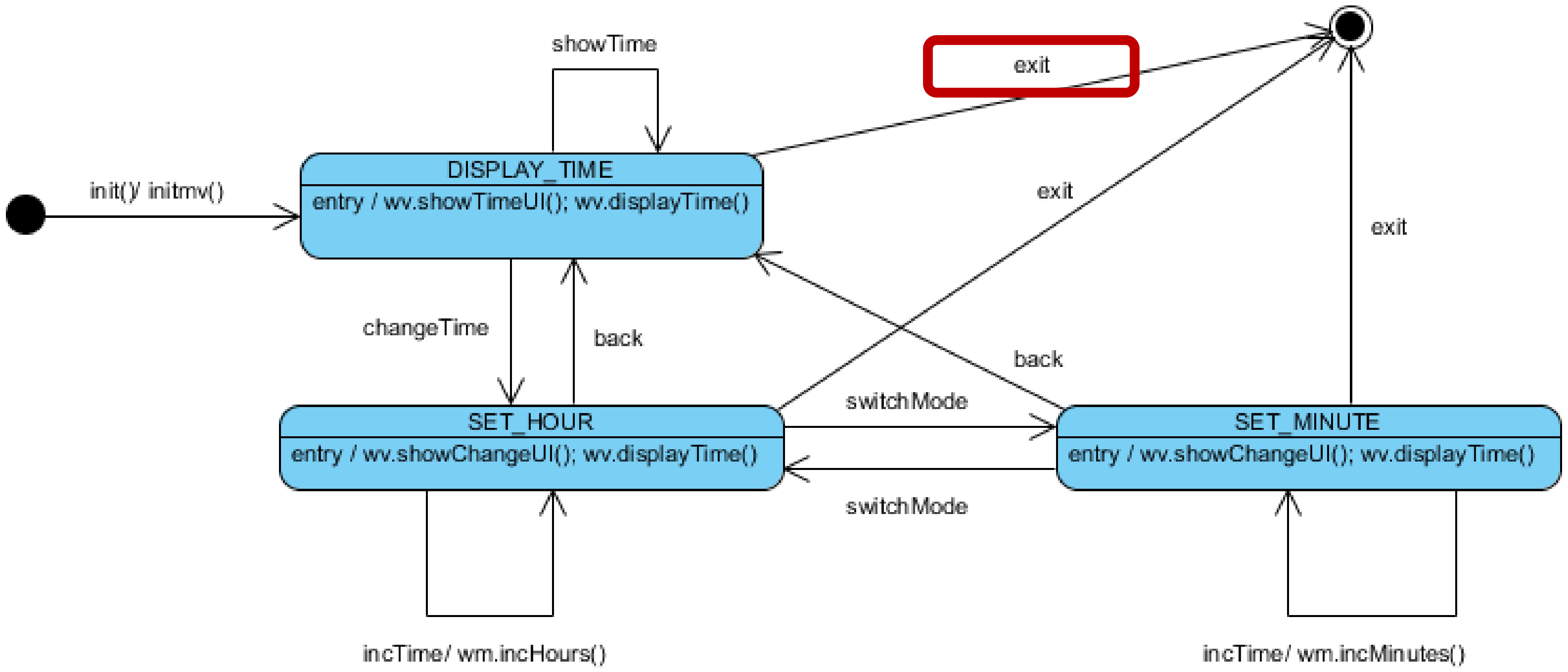


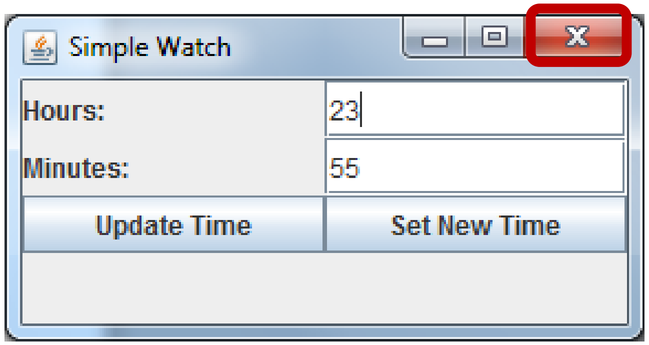


Hours:	23
Minutes:	55

Simple Watch State Chart

Graz University of Technology





Hours: 23

Minutes: 55

UI Development & JUnit Tests

Graz University of Technology

- UI Development in Java Swing: https://youtu.be/GS1Rj0wIOIk

Graz University of Technology

Thanks!

ase.ist.tugraz.at
www.felfernig.eu