BORG CALENDAR M4

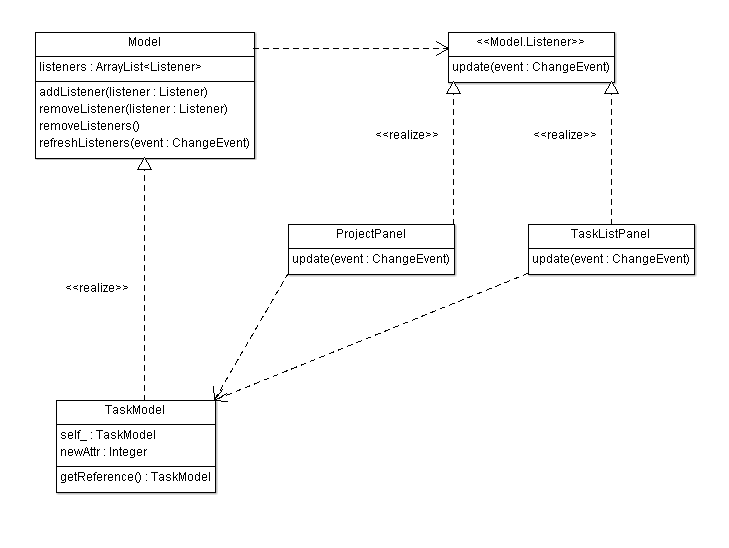
Student Name and Number

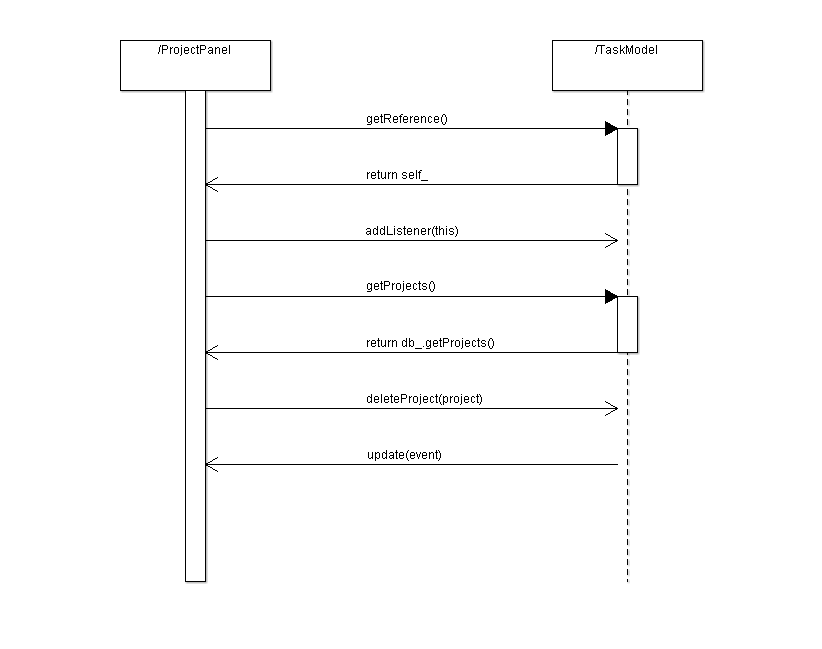
Hamid Shahrestani (9729747)  
Anjaneyulu Bodepudi (5973775)   
Manouchehr Azizi (5232287)   
Viet Hung Nguyen (9816240)

# Identify Substantial Patterns

# MVC Pattern (Viet Hung Nguyen)

MVC design pattern [3] is implemented in BORG to separate between models and UI. However it combines view and control into a UI class. A UI class get data from a model and display data to users. From user interaction, UI class manipulates its model. Then the model will then notify all listening UI about its changes.



Observer pattern is used to allow views getting notified by model changes [1].

Singleton pattern is used to allow UI classes have access to models [2].

The way of applying MVC design pattern in BORG with combined view and control is similar to the Document View pattern in MFC framework from Microsoft [4].

Using MVC in BORG allow separating between models and UI classes so that we can easily write unit tests for model classes and protect models from UI changes.

In general, the separation between view and control allows having different views, for example supporting different environment such as desktop, web browsers, mobile browsers etc., with the same control logic and allows testing controller’s logic independent with views. In BORG, view and controller are combined. So testing control logic would not be easy for BORG.

*Tools for reverse engineering:*

* AgroUML: a tool to draw class diagrams
* Intellij Idea: an IDE with strong support for navigating around the code.

*Code snippets:*

public abstract class Model {

private ArrayList<Listener> listeners;

public void addListener(Listener listener) {…}

protected void refreshListeners(ChangeEvent event) {…}

public void removeListener(Listener listener) {…}

protected void removeListeners() {…}

protected void refreshListeners(ChangeEvent event) {…}

}

public interface Listener {

public abstract void update(ChangeEvent event);

}

public class ProjectPanel … implements Model.Listener {

public void update(ChangeEvent event) {

refresh();

}

}

public class TaskModel extends Model … {

static private TaskModel self\_ = new TaskModel();

private TaskModel() {…}

static public TaskModel getReference() {

return (self\_);

}

}

# References

[1] Observer design pattern. <http://sourcemaking.com/design_patterns/observer>

[2] Singleton design pattern. <http://sourcemaking.com/design_patterns/singleton>

[3] Model View Control design patter. <http://en.wikipedia.org/wiki/Model-view-controller>

[4] MFC framework. <http://msdn.microsoft.com/en-ca/library/k9kb0kba.aspx>

# Implement a refactoring

In TaskModel class, we notice that its two methods: daysBetween() and daysLeft() is not responsible of TaskModel itself, and there is another class, DateUtil, that seems to be more suitable to contain those methods. So we decide to do this refactoring: moving daysBetween() and daysLeft() from TaskModel to DateUtil.

The patch file is patchset.patch on github at https://github.com/nvhung/soen6471/

**Patchset 0/5**: Move TaskModel.daysBetween() and TaskModel.daysLeft() to DateUtil

Commit 19df2def2e13630945b07a833c084e63614335ed

There are 5 steps to do so:

- Step 1: Copy TaskModel.daysBetween() and TaskModel.daysLeft() to DateUtil

- Step 2: Replace usages of TaskModel.daysBetween() by DateUtil.daysBetween()

- Step 3: Remove TaskModel.daysBetween()

- Step 4: Replace usages of TaskModel.daysLeft() by DateUtil.daysLeft()

- Step 5: Remove TaskModel.daysLeft()

**Patchset 1/5:** Copy TaskModel.daysBetween() and TaskModel.daysLeft() to DateUtil

Commit 88cab62dd16b688c814fb2836082e2acf2b4ebd4

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/common/DateUtil.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/common/DateUtil.java

index 0880b2d..8b585c5 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/common/DateUtil.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/common/DateUtil.java

@@ -130,5 +130,74 @@ public class DateUtil {

return hoursString + minutesString;

}

+ /\*\*

+ \* determine the number fo days between two dates

+ \*

+ \* @param start

+ \* the first date

+ \* @param dd

+ \* the later date

+ \*

+ \* @return the int

+ \*/

+ public static int daysBetween(Date start, Date dd) {

+

+ if (dd == null)

+ return 0;

+ Calendar startcal = new GregorianCalendar();

+ Calendar dcal = new GregorianCalendar();

+ dcal.setTime(dd);

+ startcal.setTime(start);

+

+ // find days left

+ int days = 0;

+ if (dcal.get(Calendar.YEAR) == startcal.get(Calendar.YEAR)) {

+ days = dcal.get(Calendar.DAY\_OF\_YEAR)

+ - startcal.get(Calendar.DAY\_OF\_YEAR);

+ } else {

+ days = new Long((dd.getTime() - startcal.getTime().getTime())

+ / (1000 \* 60 \* 60 \* 24)).intValue();

+ }

+

+ // if due date is past, set days left to 0

+ // negative days are silly

+ if (days < 0)

+ days = 0;

+ return days;

+ }

+

+ /\*\*

+ \* return the number of days left before a given date.

+ \*

+ \* @param dd

+ \* the date

+ \*

+ \* @return the number of days left

+ \*/

+ public static int daysLeft(Date dd) {

+

+ if (dd == null)

+ return 0;

+ Calendar today = new GregorianCalendar();

+ Calendar dcal = new GregorianCalendar();

+ dcal.setTime(dd);

+

+ // find days left

+ int days = 0;

+ if (dcal.get(Calendar.YEAR) == today.get(Calendar.YEAR)) {

+ days = dcal.get(Calendar.DAY\_OF\_YEAR)

+ - today.get(Calendar.DAY\_OF\_YEAR);

+ } else {

+ days = new Long((dd.getTime() - today.getTime().getTime())

+ / (1000 \* 60 \* 60 \* 24)).intValue();

+ }

+

+ // if due date is past, set days left to 0

+ // negative days are silly

+ if (days < 0)

+ days = 0;

+ return days;

+ }

+

}

**Patchset 2/5:** Replace usages of TaskModel.daysBetween() by DateUtil.daysBetween()

Commit d6b407565898daeec389e3166d9d1207ee0175ea

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

index c389189..a2f0ebc 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

@@ -44,10 +44,7 @@ import javax.swing.ListSelectionModel;

import javax.swing.event.TableModelEvent;

import javax.swing.table.TableCellRenderer;

-import net.sf.borg.common.Errmsg;

-import net.sf.borg.common.PrefName;

-import net.sf.borg.common.Prefs;

-import net.sf.borg.common.Resource;

+import net.sf.borg.common.\*;

import net.sf.borg.model.CategoryModel;

import net.sf.borg.model.Model;

import net.sf.borg.model.Model.ChangeEvent;

@@ -643,8 +640,8 @@ class TaskListPanel extends JPanel implements Model.Listener {

// duration

if (task.getDueDate() != null) {

- ro[8] = new Integer(TaskModel.daysBetween(task

- .getStartDate(), task.getDueDate()));

+ ro[8] = new Integer(DateUtil.daysBetween(task

+ .getStartDate(), task.getDueDate()));

} else {

ro[8] = null;

}

@@ -661,7 +658,7 @@ class TaskListPanel extends JPanel implements Model.Listener {

if (end == null) {

ro[9] = null;

} else {

- ro[9] = new Integer(TaskModel.daysBetween(task

+ ro[9] = new Integer(DateUtil.daysBetween(task

.getStartDate(), end));

}

@@ -729,7 +726,7 @@ class TaskListPanel extends JPanel implements Model.Listener {

ro[7] = subtask.getDueDate();

if (subtask.getDueDate() != null) {

- ro[8] = new Integer(TaskModel.daysBetween(subtask

+ ro[8] = new Integer(DateUtil.daysBetween(subtask

.getStartDate(), subtask.getDueDate()));

} else {

ro[8] = null;

@@ -746,7 +743,7 @@ class TaskListPanel extends JPanel implements Model.Listener {

if (end == null) {

ro[9] = null;

} else {

- ro[9] = new Integer(TaskModel.daysBetween(subtask

+ ro[9] = new Integer(DateUtil.daysBetween(subtask

.getStartDate(), end));

}

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

index 82afefc..a74f2e3 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

@@ -1346,7 +1346,7 @@ public class TaskView extends DockableView {

subtask.getDescription(),

subtask.getStartDate(),

subtask.getDueDate(),

- subtask.getDueDate() != null ? new Integer(TaskModel

+ subtask.getDueDate() != null ? new Integer(DateUtil

.daysBetween(subtask.getStartDate(), subtask

.getDueDate())) : null,

subtask.getDueDate() != null ? new Integer(TaskModel

**Patchset 3/5:** Remove TaskModel.daysBetween()

Commit 25b093018d557c48b226798ee6ee1493db4580d2

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

index 24f4508..332fedc 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

@@ -1073,10 +1073,10 @@ public class TaskModel extends Model implements Model.Listener, CategorySource,

/\*\*

\* return the number of days left before a given date.

- \*

+ \*

\* @param dd

\* the date

- \*

+ \*

\* @return the number of days left

\*/

public static int daysLeft(Date dd) {

@@ -1103,7 +1103,7 @@ public class TaskModel extends Model implements Model.Listener, CategorySource,

days = 0;

return days;

}

-

+

public static final int NO\_DAYS\_VALUE = 9999999;

/\*\*

\* return the days left to complete the next due item for a task

@@ -1131,42 +1131,6 @@ public class TaskModel extends Model implements Model.Listener, CategorySource,

}

- /\*\*

- \* determine the number fo days between two dates

- \*

- \* @param start

- \* the first date

- \* @param dd

- \* the later date

- \*

- \* @return the int

- \*/

- public static int daysBetween(Date start, Date dd) {

-

- if (dd == null)

- return 0;

- Calendar startcal = new GregorianCalendar();

- Calendar dcal = new GregorianCalendar();

- dcal.setTime(dd);

- startcal.setTime(start);

-

- // find days left

- int days = 0;

- if (dcal.get(Calendar.YEAR) == startcal.get(Calendar.YEAR)) {

- days = dcal.get(Calendar.DAY\_OF\_YEAR)

- - startcal.get(Calendar.DAY\_OF\_YEAR);

- } else {

- days = new Long((dd.getTime() - startcal.getTime().getTime())

- / (1000 \* 60 \* 60 \* 24)).intValue();

- }

-

- // if due date is past, set days left to 0

- // negative days are silly

- if (days < 0)

- days = 0;

- return days;

- }

-

**Patchset 4/5:** Replace usages of TaskModel.daysLeft() by DateUtil.daysLeft()

Commit cb79d02e2c783245f9e77b14e179b59a2259bbd6

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

index 332fedc..6c91c6b 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

@@ -1117,14 +1117,14 @@ public class TaskModel extends Model implements Model.Listener, CategorySource,

if( isClosed(t)) return daysLeft;

if( t.getDueDate() != null )

- daysLeft = daysLeft(t.getDueDate());

+ daysLeft = DateUtil.daysLeft(t.getDueDate());

for( Subtask st : getSubTasks(t.getKey()))

{

if( st.getCloseDate() != null ) continue;

if( st.getDueDate() != null )

- daysLeft = Math.min(daysLeft(st.getDueDate()), daysLeft);

+ daysLeft = Math.min(DateUtil.daysLeft(st.getDueDate()), daysLeft);

}

return daysLeft;

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectPanel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectPanel.java

index f31a99d..c09f189 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectPanel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectPanel.java

@@ -39,6 +39,7 @@ import javax.swing.ListSelectionModel;

import javax.swing.event.TableModelEvent;

import javax.swing.table.TableCellRenderer;

+import net.sf.borg.common.DateUtil;

import net.sf.borg.common.Errmsg;

import net.sf.borg.common.Resource;

import net.sf.borg.common.Warning;

@@ -558,7 +559,7 @@ public class ProjectPanel extends JPanel implements Model.Listener {

ro[7] = new Integer(MAGIC\_NO\_DUE\_DATE);

else {

Date dd = (Date) ro[4];

- ro[7] = new Integer(TaskModel.daysLeft(dd));

+ ro[7] = new Integer(DateUtil.daysLeft(dd));

}

// strip newlines from the description

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectView.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectView.java

index 3c179db..29730af 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectView.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/ProjectView.java

@@ -506,7 +506,7 @@ public class ProjectView extends DockableView {

gc2.setTime(dd);

startDateChooser.setCalendar(gc2);

- int daysleft = TaskModel.daysLeft(p.getDueDate());

+ int daysleft = DateUtil.daysLeft(p.getDueDate());

daysLeftText.setText(Integer.toString(daysleft));

String cat = p.getCategory();

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

index a2f0ebc..51e3c9a 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskListPanel.java

@@ -667,7 +667,7 @@ class TaskListPanel extends JPanel implements Model.Listener {

ro[4] = null;

else {

Date dd = (Date) ro[7];

- ro[4] = new Integer(TaskModel.daysLeft(dd));

+ ro[4] = new Integer(DateUtil.daysLeft(dd));

}

// strip newlines from the description

@@ -752,7 +752,7 @@ class TaskListPanel extends JPanel implements Model.Listener {

ro[4] = null;

else {

Date dd = (Date) ro[7];

- ro[4] = new Integer(TaskModel.daysLeft(dd));

+ ro[4] = new Integer(DateUtil.daysLeft(dd));

}

ro[5] = subtask.getDescription();

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

index a74f2e3..1aca6cf 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/ui/task/TaskView.java

@@ -169,7 +169,7 @@ public class TaskView extends DockableView {

/\*

\* color the due date background based on days left

\*/

- int days = TaskModel.daysLeft(dd);

+ int days = DateUtil.daysLeft(dd);

if (!isSelected) {

// yellow alert

if (days < Prefs.getIntPref(PrefName.YELLOW\_DAYS))

@@ -1314,7 +1314,7 @@ public class TaskView extends DockableView {

closeDate.setText(DateFormat.getDateInstance(DateFormat.MEDIUM)

.format(cd));

- int daysleft = TaskModel.daysLeft(task.getDueDate());

+ int daysleft = DateUtil.daysLeft(task.getDueDate());

daysLeftText.setText(Integer.toString(daysleft));

String cat = task.getCategory();

@@ -1349,7 +1349,7 @@ public class TaskView extends DockableView {

subtask.getDueDate() != null ? new Integer(DateUtil

.daysBetween(subtask.getStartDate(), subtask

.getDueDate())) : null,

- subtask.getDueDate() != null ? new Integer(TaskModel

+ subtask.getDueDate() != null ? new Integer(DateUtil

.daysLeft(subtask.getDueDate())) : null,

subtask.getCloseDate() };

**Patchset 5/5:** Remove TaskModel.daysLeft()

Commit 22bbbbf65d6335ca56f750c95529e74b6af93c29

diff --git a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

index 6c91c6b..b9baa7a 100644

--- a/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

+++ b/Source Code/BORG/borg\_src/BORGCalendar/src/net/sf/borg/model/TaskModel.java

@@ -1071,39 +1071,6 @@ public class TaskModel extends Model implements Model.Listener, CategorySource,

return db\_.getLogs();

}

- /\*\*

- \* return the number of days left before a given date.

- \*

- \* @param dd

- \* the date

- \*

- \* @return the number of days left

- \*/

- public static int daysLeft(Date dd) {

-

- if (dd == null)

- return 0;

- Calendar today = new GregorianCalendar();

- Calendar dcal = new GregorianCalendar();

- dcal.setTime(dd);

-

- // find days left

- int days = 0;

- if (dcal.get(Calendar.YEAR) == today.get(Calendar.YEAR)) {

- days = dcal.get(Calendar.DAY\_OF\_YEAR)

- - today.get(Calendar.DAY\_OF\_YEAR);

- } else {

- days = new Long((dd.getTime() - today.getTime().getTime())

- / (1000 \* 60 \* 60 \* 24)).intValue();

- }

-

- // if due date is past, set days left to 0

- // negative days are silly

- if (days < 0)

- days = 0;

- return days;

- }

-

public static final int NO\_DAYS\_VALUE = 9999999;

/\*\*

\* return the days left to complete the next due item for a task