

TALLINN UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF SOFTWARE SCIENCE

Coursework on Contracts

Homework in subject "Software assurance" (ITI8610),
Module III, Group Assignment

Authors:

Olga Andrejeva

Madis Kõosaar

Pavel Lavrešin

Kevin Laanemägi

Hendrik Kivi

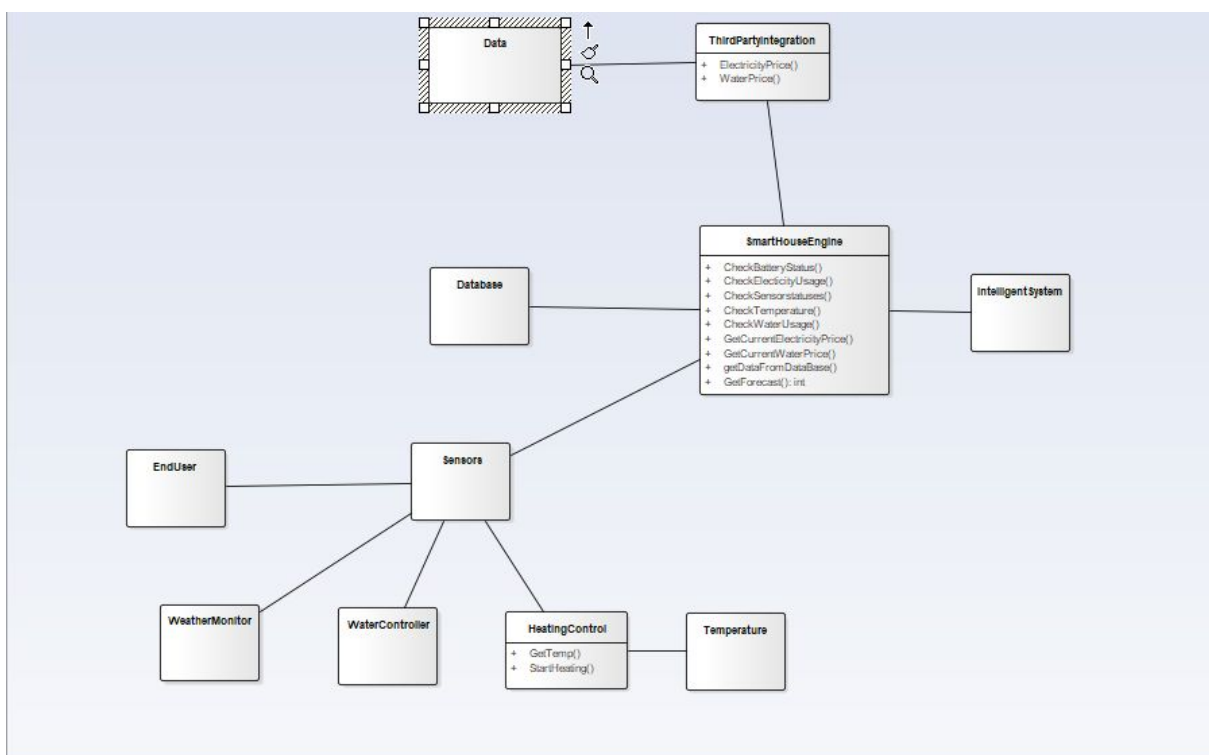
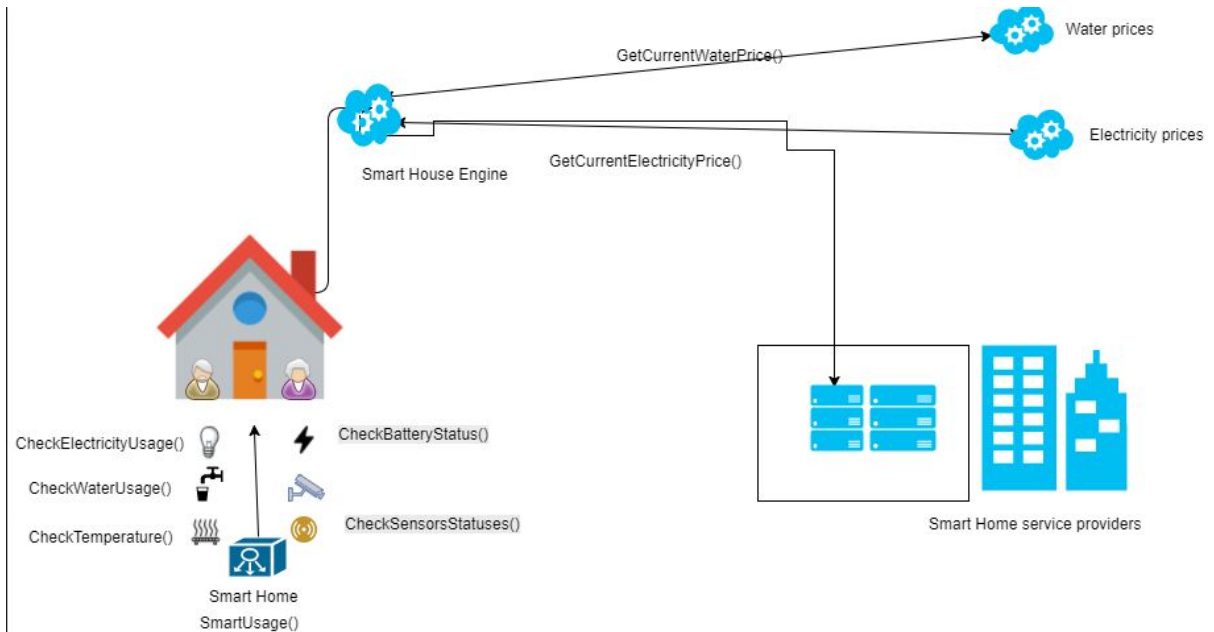
Architecture diagram	3
Communication diagram	4
Source code	5
Contract assertions	5
Quality attributes	6
Contracts by views and provide multi-view contracts	7
OpenJML Type-checking run	7

Architecture diagram

Architecture overview is presented below and contains architectural entities (actors, components, parties, objects, etc).

[Drawing #1](#)

[Drawing #2](#)

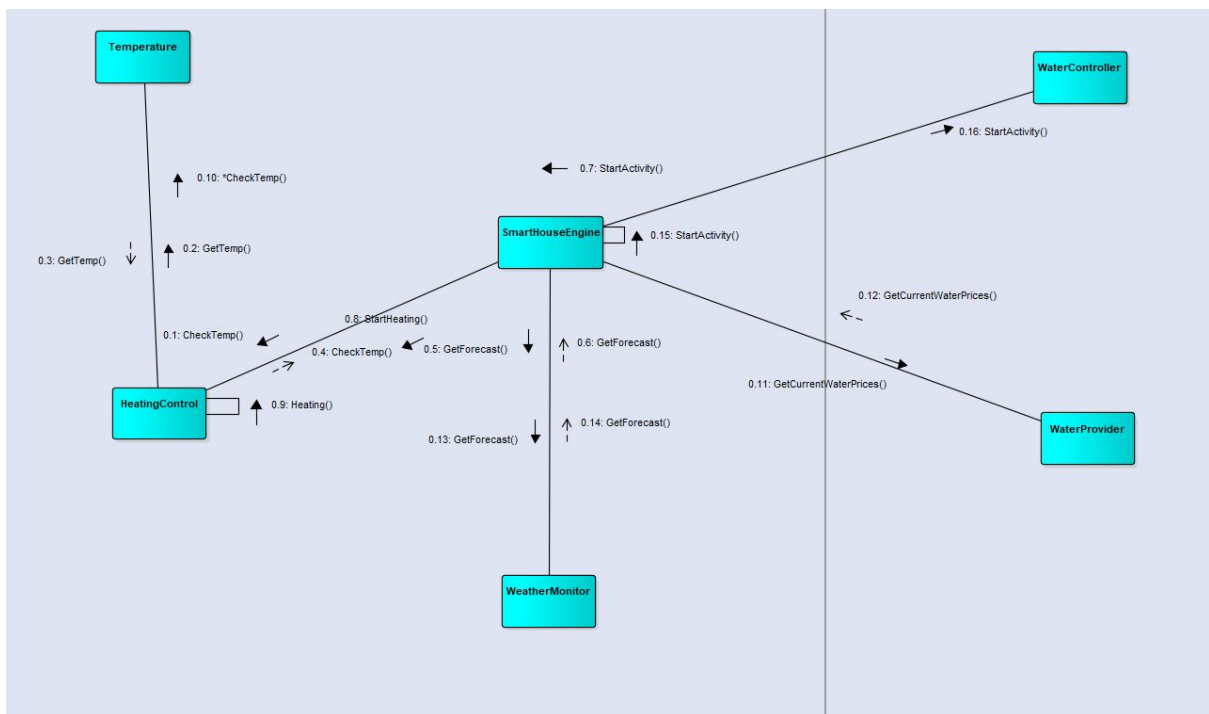
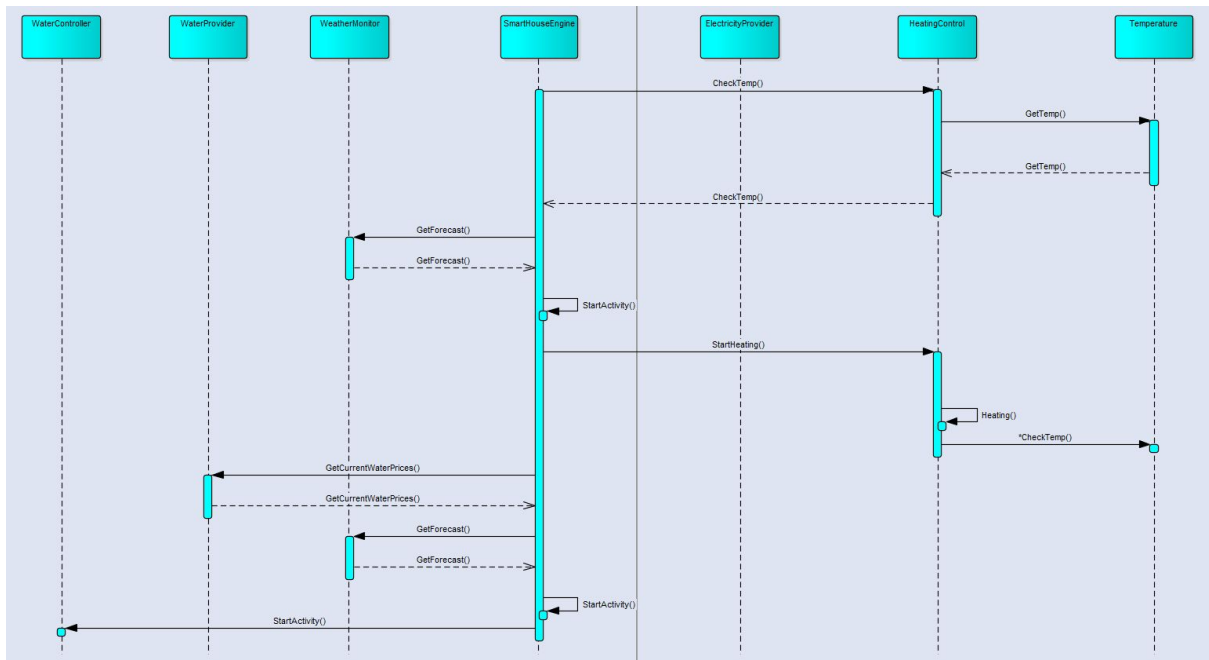


Communication diagram

Communication and interaction between entities (actors, components, parties, objects, etc) are presented below.

[Drawing #1](#)

[Drawing #2](#)



Source code

Repository with source code of Java app command-line and OpenJML contracts is defined below.

<https://github.com/plavreshin/smart-home>

Contract assertions

Most of domain entities are defined with required contract assertions:

<https://github.com/plavreshin/smart-home/tree/master/src/main/java/ee/ttu/study/domain>

An example of [ElectricalDevice.java](#) attached below:

```
package ee.ttu.study.domain;

public class ElectricalDevice {
    private /*@ spec_public @*/ final String name;
    private /*@ spec_public @*/ final Integer peakPower;

    /*@ requires name != null;
    /*@ requires peakPower != null;
    public ElectricalDevice(String name, Integer peakPower) {
        this.name = name;
        this.peakPower = peakPower;
    }

    /*@ ensures \result == name;
    public String getName() {
        return name;
    }

    /*@ ensures \result == peakPower;
    /*@ pure @*/
    public Integer getPeakPower() {
        return peakPower;
    }
}
```

Quality attributes

Some of modules are defined with required quality attributes:

<https://github.com/plavreshin/smart-home/blob/master/src/main/java/ee/ttu/study/module/>

An example of [HeatingControl.java](#) attached below:

```
public class HeatingControl implements Module {
    private static final Logger log =
LoggerFactory.getLogger(HeatingControl.class);
    private List<Room> monitoredRooms;

    public HeatingControl(List<Room> monitoredRooms) {
        EventBusEngine.register(this);
        this.monitoredRooms = monitoredRooms;
    }

    /**
     * @requires event != null;
     */
    @Subscribe
    public void onPriceChange(final PriceChangeEvent event) {
        if (isOutdated(event)) {
            log.info("Received priceChangeEvent: " + event);
            monitoredRooms.forEach(x -> x.calculateUsage(event.getPrice()));
        }
    }

    /**
     * Timing
     * @requires event != null;
     * @requires
     * Duration.between(Instant.now(),event.timestamp).toMillis() > 1000;
     * @
     * @also
     * @
     * @requires event != null;
     * @requires
     * Duration.between(Instant.now(),event.timestamp).toMillis() < 1000;
     */
    public boolean isOutdated(PriceChangeEvent event) {
        return Duration.between(Instant.now(),
event.getTimestamp()).toMillis() > 1000;
    }

    /**
     * @requires event != null;
     */
}
```

```

@Subscribe
public void onTemperatureChange(final TemperatureChangeEvent event) {
    log.info("Received temperatureChangeEvent: " + event);
    monitoredRooms.forEach(x ->
x.updateTemperature(event.getTemperature()));
}

@Override
public void initialize() {

}
}

```

Contracts by views and provide multi-view contracts

Some of modules are defined with `also` keyword to support contract views.:

<https://github.com/plavreshin/smart-home/blob/master/src/main/java/ee/ttu/study/module/>

An example of [HeatingControl.java](#) attached below:

```

/* Timing */
    // @ requires event != null;
    // @ requires
Duration.between(Instant.now(),event.timestamp).toMillis() > 1000;
    // @
    // @also
    // @
    // @ requires event != null;
    // @ requires
Duration.between(Instant.now(),event.timestamp).toMillis() < 1000;
    public boolean isOutdated(PriceChangeEvent event) {
        return Duration.between(Instant.now(),
event.getTimestamp()).toMillis() > 1000;
    }
}

```

OpenJML Type-checking run

An output of OpenJML type-checking is attached below and it has not revealed any significant issues despite few warnings of **non-pure** method usages.

```

[loading
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jd

```



```
Duration.between(Instant.now(),event.timestamp).toMillis() < 1000;
```

^

```
[loading  
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/lib/ct.sym(META-INF/sym/rt.jar/java/util/concurrent/ScheduledFuture.class)]]
```

```
[loading  
ZipFileIndexFileObject[/Users/pavel/.gradle/caches/modules-2/files-2.1/com.google.guava/guava/23.5-jre/e9ce4989adf6092a3dab6152860e93d989e8cf88/guava-23.5-jre.jar(com/google/common/eventbus/SubscriberExceptionHandler.class)]]
```

```
[loading  
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/lib/ct.sym(META-INF/sym/rt.jar/java/lang/reflect/TypeVariable.class)]]
```

```
/Users/pavel/eclipse/java-oxygen/Eclipse.app/Contents/Eclipse/configuration/org.eclipse.osgi/504/0/.cp/java5/java/lang/reflect/Method.jml:53:  
warning: A non-pure method is being called where it is not permitted:  
java.lang.reflect.Method.getModifiers()  
    //+@ requires !Modifier.isStatic(getModifiers()) ==> obj != null;
```

^

```
[loading  
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/lib/ct.sym(META-INF/sym/rt.jar/java/lang/reflect/Modifier.class)]]
```

```
[loading  
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/lib/ct.sym(META-INF/sym/rt.jar/java/lang/Byte$ByteCache.class)]]
```

```
[loading  
ZipFileIndexFileObject[/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/lib/ct.sym(META-INF/sym/rt.jar/java/lang/Short$ShortCache.class)]]
```

Note:

```
/Users/pavel/eclipse/java-oxygen/Eclipse.app/Contents/Eclipse/configuration/org.eclipse.osgi/504/0/.cp/java8/java/util/Random.jml uses internal proprietary API that may be removed in a future release.
```

Note: Recompile with `-Xlint:sunapi` **for** details.

[1.40] Completed