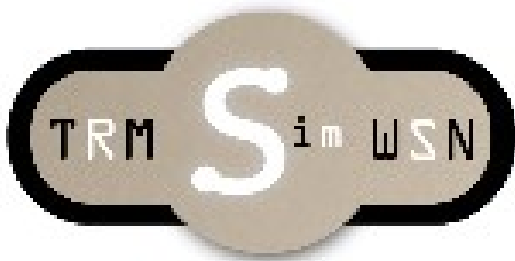




Implementing and Integrating a new Trust and/or Reputation Model in TRMSim-WSN



Félix Gómez Mármol

<http://ants.dif.um.es/~felixgm>

felixgm@um.es

8th June 2009

This document describes the steps to be followed in order to effectively implement and integrate a new trust and/or reputation model for distributed networks within the simulator [TRMSim-WSN \[1\]](#).

Hence, if a fictitious trust and/or reputation model called **TemplateTRM** was to be implemented and integrated in [TRMSim-WSN version 0.3](#) and beyond, the next steps should be carried out:

1. Create a folder inside `trmodels` folder named `templatetrm`.
2. Create a parameters file inside folder `templatetrm`, called `TemplateTRMparameters.txt` (download it [here](#)) with a content similar to:

```
#####
# TemplateTRM parameters file
#####
parameter1Name=parameter1Value
parameter2Name=parameter2Value
...
```

3. Create a package called `es.ants.felixgm.trmsim_wsn.trm.templatetrm` (from now on, 'subpackage `templatetrm`').

4. Create a subclass of class `TRMParameters` called `TemplateTRM_Parameters` in the subpackage `templatetrm` with the following content (at least):



```
public class TemplateTRM_Parameters extends TRMParameters {
    public static final String defaultParametersFileName =
"trmodels"+File.separator+"templatetrm"+File.separator+"Templat
eTRMparameters.txt";

    public TemplateTRM_Parameters() {
        super();
    }

    public TemplateTRM_Parameters(String fileName) throws
Exception {
        super(fileName);
    }

    public String toString() {
        return "";
    }
}
```

5. Within the package `es.ants.felixgm.trmsim_wsn.gui.parameterpanels` create a subclass of class `TRMParametersPanel` called `TemplateTRM_ParametersPanel` with the following content (at least):

```
public class TemplateTRM_ParametersPanel extends
TRMParametersPanel {

    public TemplateTRM_ParametersPanel() {
        initComponents();
    }

    public TRMParameters get_TRMParameters() {
        TemplateTRM_Parameters templateTRMParameters =
            new TemplateTRM_Parameters();

        return templateTRMParameters;
    }

    public void set_TRMParameters(TRMParameters trmParameters){
    }

    public void setEnabled(boolean enabled) {
    }
}
```



6. Create a subclass of class `Sensor` called `TemplateTRM_Sensor` in the subpackage `templatetrm` with the following content (at least):

```
public class TemplateTRM_Sensor extends Sensor {  
  
    public TemplateTRM_Sensor () {  
        super();  
    }  
  
    public TemplateTRM_Sensor(int id, double x, double y) {  
        super(id,x,y);  
    }  
  
    public void reset() {  
    }  
}
```

7. Create a subclass of class `Network` called `TemplateTRM_Network` in the subpackage `templatetrm` with the following content (at least):

```
public class TemplateTRM_Network extends Network {  
  
    public TemplateTRM_Network(  
        int numSensors,  
        double probClients,  
        double rangeFactor,  
        Collection<Double> probServices,  
        Collection<Double> probGoodness,  
        Collection<Service> services) {  
        super(numSensors,      probClients,      rangeFactor,  
probServices, probGoodness, services);  
        reset();  
    }  
  
    public TemplateTRM_Network(String xmlFilePath) throws  
Exception {  
        super(xmlFilePath);  
        reset();  
    }  
  
    public Sensor newSensor(){  
        return new TemplateTRM_Sensor();  
    }  
}
```



```
    public Sensor newSensor(int id, double x, double y) {  
        return new TemplateTRM_Sensor(id,x,y);  
    }  
}
```

8. Create a subclass of class TRModel_WSN called TemplateTRM in the subpackage `templatetrm` (it is important that both the model and the package have the same name, ignoring upper and lowercase letters, in order to be automatically detectable by the simulator) with the following content (at least):

```
public class TemplateTRM extends TRModel_WSN {  
  
    public TemplateTRM(TemplateTRM_Parameters  
templateTRM_parameters) {  
        super(templateTRM_parameters);  
    }  
  
    public static String get_name() { return "TemplateTRM"; }  
  
    public GatheredInformation gatherInformation(Sensor client,  
Service service) {  
        return null;  
    }  
  
    public Vector<Sensor> scoreAndRanking(Sensor client,  
GatheredInformation gi) {  
        return null;  
    }  
  
    public Outcome performTransaction(Vector<Sensor> path,  
Service service) {  
        return null;  
    }  
  
    public Outcome reward(Vector<Sensor> path, Outcome outcome)  
{  
        return outcome;  
    }  
  
    public Outcome punish(Vector<Sensor> path, Outcome outcome)  
{  
        return outcome;  
    }  
}
```



```
public Network generateRandomNetwork(  
    int numSensors,  
    double probClients,  
    double rangeFactor,  
    Collection<Double> probServices,  
    Collection<Double> probGoodness,  
    Collection<Service> services) {  
    return new  
TemplateTRM_Network(numSensors,probClients,rangeFactor,probServ  
ices,probGoodness,services);  
}  
  
public Network loadCurrentNetwork(String fileName) throws  
Exception {  
    return new TemplateTRM_Network(fileName);  
}  
}
```

9. Place all the newly created classes in the folder `src/es/ants/felixgm/trmsim_wsn/trm/templatetrm` and the simulator will recognize the new model automatically and incorporate it in the forthcoming executions.

- [1] Félix Gómez Mármol, Gregorio Martínez Pérez, "[TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks](#)", *IEEE International Conference on Communications (IEEE ICC 2009), Communication and Information Systems Security Symposium, Dresden, Germany, 14-18 June 2009*