Tutorial PHATSIM

(Part 1 of 4)

European Social Simulation Conference 2014

Authors:

Jorge J. Gómez Sanz

Pablo Campillo

{jjgomez,pabcampi}@ucm.es

Universidad Complutense de Madrid

http://grasia.fdi.ucm.es/sociaal

Before anything

Ensure the following software is installed into your computer:

- JDK 1.7. You can get this software at http://java.sun.com. Make sure JAVA_HOME environment variable is set to the home of JAVA. JRE is not enough, you need the compiler too.
- Maven 3.1.1 (or higher). You can get this at http://maven.apache.org/download.html. Make sure the M2_HOME environment variable exists and points to the Maven home
- Ant 1.9.3 (or higher). You can get this software at http://ant.apache.org. Make sure the ANT_HOME environment variable exists and that points to the ant home.
- For all the abovementioned, make sure the PATH environment variable point at the binaries of each one.

If you want to run too the Android part

- Android SDK (r21.1 or later, latest is best supported) installed, preferably with all platforms, see http://developer.android.com/sdk/index.html
- Make sure you have images for API 19 installed in your android SDK. It is required to have the **IntelAtomx86** image to permit hardware acceleration. Instructions for Android are available in the Android site (http://developer.android.com/tools/devices/emulator.html#acceleration)
- Set environment variable ANDROID_HOME to the path of your installed Android SDK and add $ANDROID_HOME/tools as well as $ANDROID_HOME/platform-tools to your $PATH. (or on Windows %ANDROID_HOME%\tools and %ANDROID_HOME %\platform-tools).
- Add binaries to environment variable PATH (PATH=$PATH:$HOME/bin: $JAVA_HOME/bin:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools: $M2_HOME/bin)

Quick start

A development at this stage involves the following set of steps:

1. Create a model of the situation
    a. In the console, type “ant edit” and hit enter
2. Run the simulation
    a. In the console, type “ant runSim1” and hit enter

   b. OBS: the simulation run is specific of each model. The pattern in general is “ant runXXX” where XXX is the name of the simulation
3. Check the results match the intended results
   a. If they do, the job is finished
   b. If it does not, modify the model and run again
4. Share your results with others. Simulations can be packaged in a jar so that others can reproduce them. They can be also be documented as HTML documents that explain the simulation model.

You can edit a model while you run, but you should not save it until the model is running. Saving the model while the system is compiling may lead to fake failures.

The quick introduction includes a simple use case where a person performs the following actions:

1. Go to the living room
2. Turn on the Tv
3. Sit down at the Sofa

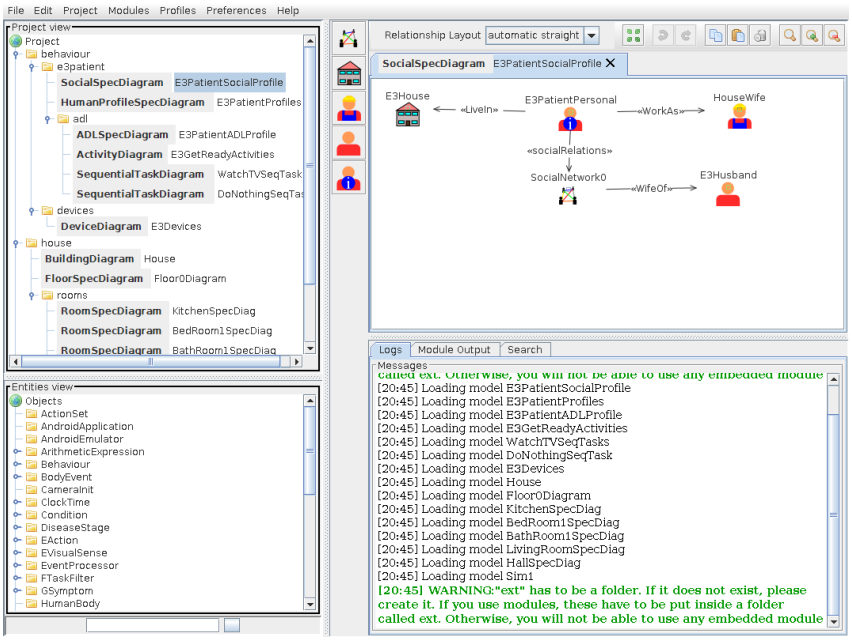
Each of the development activities is further explained in the following sections.

Editing the simulation

A model in PHAT is made of several diagrams showing aspects of the simulation. There are several diagram types. Each diagram contains interconnected entities which may appear more than once in the same diagram or across several diagrams.

Any simulation will be made of several diagrams ordered following the developer’s criteria. A developer can group diagrams in packages, which are represented with folders icons, and move them from one to another.

Models are created with the editor which the figure shows. The editor permits to browse the different diagrams in the project tree (top left). The content of each diagram is shown in the top right section by double clicking into the corresponding diagram. A dictionary of all defined entities in the current model can be seen in the bottom left section. The bottom right section shows error messages and other convenient information.



Layout of elements in each diagram is not meaningful, but the connections are. Two elements cannot be linked together in just anyway. A diagram will follow always certain grammar which is called a meta-model. The metamodel code is published at http://sourceforge.net/p/sociaal/codegit/ci/master/tree/

Editing one model implies one or more of the following actions:

- Adding a diagram. Diagrams are added by right clicking on the desired folder and choosing a diagram type
- Deleting a diagram. A diagram removal is ordered when you right click on a diagram and choose “remove package/diagram”
- Modifying a diagram. A diagram can be modified in the following ways:
    - Removing an entity from the diagram. This is done by selecting and pressing the del key
    - Removing a relationship among entities. This is done by selecting the relationship and hitting the del key
    - Adding a new entity. A new entity can be added
        - by clicking on one of the buttons on the left hand side of the diagram
        - by right clicking at the background of the diagram and choosing “Insert XXX”, where XXX is the name of one entity
    - Adding a new relationship. A new relationship can be added by finding a small rectangle at the center of the entity, left-clicking on it, and dragging towards the

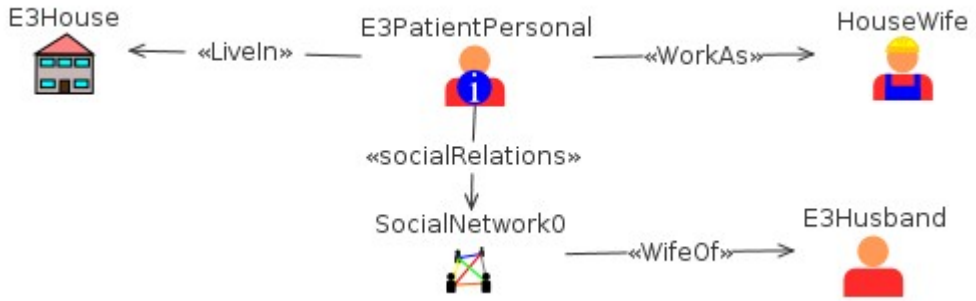
other entity with which a connection is expected. You have to drag until finding a similar central square in the target entity. At that point, just drop. A dialog may appear asking for the specific kind of relationship to use (if more than one is eligible) and the ordering of elements to be used as source and target of the relationship (if more than one order is possible).

To initiate into the modeling activities, we propose some exercises to follow.

Exercise 0-A

The exercise is about learning how use the tools to model. The tools are started from command line.

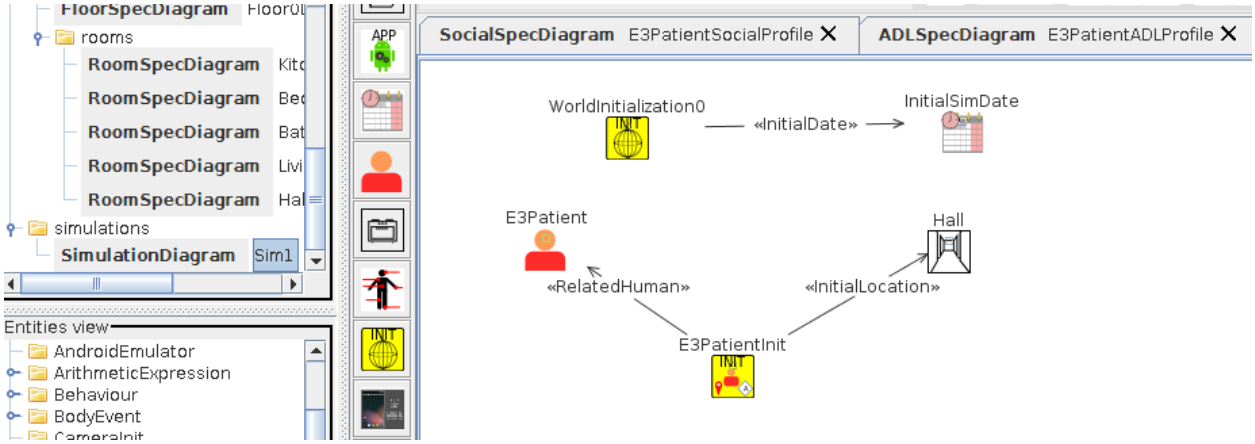
- run “ant edit”
- open the diagram “E3PatientSocialProfile”. The following should appear



- Select the “LiveIn” relationship. Hit “del”
- Reconnect E3House and E3PatientPersonal by dragging from the square in the middle of the entity E3House to the square in the middle of E3PatientPersonal

Running the simulation

The model has one distinguished diagram type which is the “SimulationDiagram”. The name of the diagram determines the name of the simulation. In this case, the name of diagram is Sim1. It configures some simulation parameters, including the actors to include. Henceforth, the name of the simulation will be Sim1



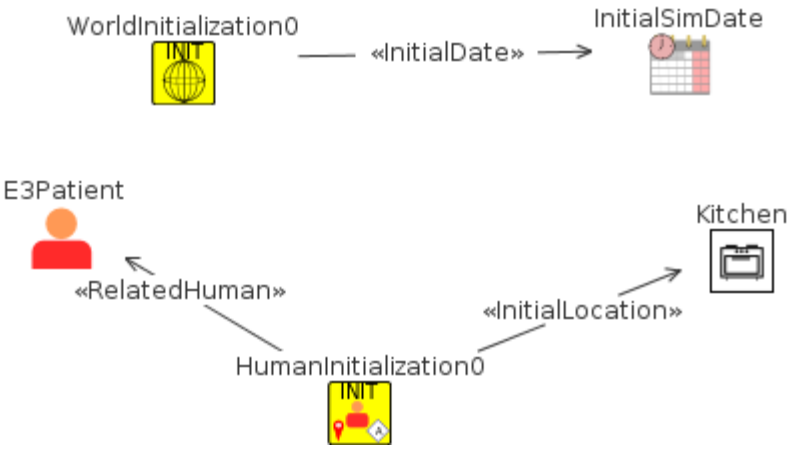
To run this simulation, type in the project folder “ant compile” and then “ant runSim1”.

Exercise 0-B

In this exercise a new simulation diagram is created. This illustrates how to handle several simulations starting from the same specification.

- Create another SimulationDiagram with name Sim2
- Go to the Sim1 diagram and select everything. Hit Ctrl+c or click in the copy icon in the toolboar above the diagram
- Go to the Sim2 diagram. Hit Ctrl+p or click in the paste icon in the toolbar above the diagram. You should have now an exact copy of the Sim1 diagram.
- Now, remove E3PatientInit and Hall
- Add another HumanInitialization entity
- Go to the Floor0Diagram and select one entity representing a place. Here, we choose Kitchen. Select the entity and copy it.
- Go to the Sim2 diagram and paste the entity.
- Connect the Kitchen entity with the HumanInitialization instance
- Open the HumanInitialization entity with a double left click, scroll down and choose “yes” in the ShowName field
- Save the file

The result should be something like this



Now, run “ant compile” and then “ant runSim2”. Try too “ant runSim1”

Sharing the simulation

Simulations can be packaged so that they can be distributed to other interested researchers. Packaging is performed over a particular simulation. To package the simulation “Sim1”, type in the console “ant packageSim1”. In general the packaging instruction will have the form “ant packageXXX” where XXX is the name of the simulation.

The result of the packaging is a jar file allocated in the target folder of your project. Go there after running the “ant packageSim1” command and find a jar with the name “initial-1.0.0-SNAPSHOT-Sim1-selfcontained.jar”. Double click over the jar from a the file explorer or write the following in a console while visiting the same folder “java -jar initial-1.0.0-SNAPSHOT-Sim1-selfcontained.jar”.

This jar is self-contained. It can be taken anywhere and run by anyone as long as this person has the appropriate java version (1.7 or above and JRE is enough). No need to have as a complex installation as one developer.

Exercise 0-C

This exercise serves to learn how to share yor simulations with others.

Package the simulation “Sim2” and try to execute it somewhere else.

Documenting the simulation

Model specifications can be documented by editing the “Description” field of each entity. These text areas can hold also pegdown code, a variant of markdown. Markdown is a quick way of adding format to the text while still using plain text editors. You can check some examples here:https://github.com/sirthias/pegdown and a more markdown specific guide here http://daringfireball.net/projects/markdown/.

To generate the documentation of the simulation, do type the following “mvn clean site”. The generated doc will be stored in the target/site folder of your project and can be viewed starting from the index.html file. The document generation can take a while, though.

Exercise 0-D

Go to some diagram and edit the text of some description field. In the example, we edit the diagram “Sim1” properties (right click over the diagram and select properties option). Try the following formats:

- A word between asterisks “*example*”
- A title level 1 “# level one title”
- A title level 2 “## level two title”
- A list of things, write this repeated into several lines “* element1”