Snoopy Computational Steering Framework
Quick Start
Version 1.0

Mostafa Herajy and Monika Heiner
– Data Structures and Software Dependability –
Institute of Computer Science
Brandenburg University of Technology
Cottbus, Germany

snoopy@informatik.tu-cottbus.de

August 20, 2013

*Please sent all questions, comments and suggestions how to improve this material to this address.
## Contents

1 Introduction  

2 Overview  
   2.1 Framework  
   2.2 Application Scenario  

3 System Requirements  
   3.1 Hardware Requirements  
   3.2 Software Requirements  

4 Installation  
   4.1 Install Snoopy under Mac OS X  
   4.2 Install SSServer under Mac OS X  
   4.3 Install Snoopy under Windows  
   4.4 Install SSServer under Windows  

5 Getting Started  
   5.1 Launching Snoopy and SSServer  
   5.2 Connecting to the Server  
      5.2.1 Configuring Connection Settings  
      5.2.2 Saving Connection Settings  
   5.3 Submit a New Model  
   5.4 Using an Existing Model  
   5.5 Replacing an Existing Model  

6 Further Reading
1 Introduction

With the advances of computing power and the proliferation of multi-core processors, it becomes essential to execute long running and computationally expensive simulations at powerful and remote computers – which enjoy high speed computational units – to profit from such precious processing resources. However, such powerful computers do not provide a direct interactive visualisation and analysis of the resulting simulation data due to either the intrinsic patch processing approach of these computers or the sharing of their resources between different users. Thus, there is a need to remotely manage and analyse the simulation output traces simultaneously while the simulation is in progress. Correspondingly, many different techniques have been proposed to overcome these limitations. Computational steering is among the elegant and promising tools that provide a tight coupling between simulation and visualisation modules of scientific models.

In this Quick Start manual we briefly discuss the use of Snoopy’s computational steering framework to simulate and interactively steer Petri nets, e.g., biochemical network models. There are a lot of important functionalities that are provided by this framework in order to facilitate the conduction of ”wet-lab” experiments. If you are wondering about what these tools can add to your work flow, it might be worth reading the following paragraphs.

The following are some aspects of what you can do using the Snoopy steering framework:

- **Remotely run and control a simulation**
  You can run your simulation at a remote computer. This feature allows the adaptation of high computational power machines from a local computer. The simulation will run at a server machine while the visualization of the results is done at a different machine, serving as a GUI client.

- **Execute the same model using different simulation algorithms**
  Sometimes it is useful to study a model in different paradigms, i.e., stochastic, continuous, or hybrid. In this framework, the same model definition can be executed with different simulation algorithms.

- **Manage concurrently different models with possibly different simulators**
  Different models can be simultaneously executed at the server side. Each model is assigned a separate simulator and can be executed independently from other
• **Define different views to explore the simulation results**
  Views provide a quick means to explore model results from different perspectives. Each view is defined by a set of curves and their associated attributes. Different views can be defined for the same model.

• **Explore on the fly your running models**
  Using the steering graphical user interface (Steering GUI), you can easily navigate among different models. Besides, the list of running models at the server side can be refreshed if another user adds a new model to the server.

• **Steer the simulation parameters while the simulation is running**
  The main goal of Snoopy steering framework is to enable users to interact with their models during simulations. Users can change model parameters as well as current marking and immediately monitor the system’s response for such changes. This is an useful tool since a user is allowed to ask ”what-if” questions.

• **Control the simulation speed**
  The simulation speed can be set to an appropriate level to facilitate the interaction with a running model during its execution. This feature is important if the simulation parameters are allowed to be changed while the simulation is running.

• **Connect to your simulation at any time from whatever place**
  The overall organization of this framework is flexible to let users connect/disconnect to/from running models without affecting their execution. Moreover, you can connect to your models from different places, for example from your office or from your home.

• **Collaborate with other people while executing model dynamics**
  More than one user are permitted to connect to the same models. Users can collaborate in executing and steering a running simulation. This feature might promote the sharing of knowledge between different users with different backgrounds.

• **Platform-independent implementation**
  The core communication library is written in standard C++ and therefore it can run on different platforms among them are windows, Mac OS X, and Linux. Moreover, the client and server does not need to run under the same platform.
2 Overview

In this section we give a general overview of the Snoopy steering framework to help understanding the high level organization before going into details.

2.1 Framework

Figure 1 presents the general architecture of the Snoopy steering framework. Its main components are: the steering server, the steering graphical user interface, the steering application programming interface (APIs), and the internal and external simulators. These interdependent ingredients enable users not only to run their biochemical network models and obtain results, but also to share, distribute and interactively steer them. Additionally, users do not need to wait until the simulation ends in order to discover potentially incorrect results. Instead, using this framework, errors could be discovered early and be immediately corrected during the simulation and, if necessary, the simulation could be restarted using the current setting. Subsequently, the overall time required to carry out dry-lab experiments will substantially decrease.

The main component of the architecture is the steering server. It is the central manager of the model data and communication traffic between the different framework components. It is a multi-user, multi-model, multi-simulator, and multi-threaded server. Inside the server, data is organised in terms of individual models which are defined by means of Petri nets.

The steering graphical user interface is the user’s entry point to the overall architecture. Through it, the user can monitor and steer the simulation output and the corresponding key parameters, respectively. Users can flexibly connect and disconnect from their local machines to the available steering servers and view the currently running models. Model dynamics are produced using either an internal or an external simulator. Internal simulators are implemented inside the server which currently supports deterministic, stochastic, and hybrid algorithms, while external simulators are defined by the user and dynamically linked to the running server.

The steering application programming interfaces (APIs) are used to incorporate external simulators into the steering server. Additional responsibility of the API library is to facilitate the connections between the different framework components. More specifically, it is used to carry out the communication between the steering GUI and the steering server.

Finally, this framework permits the simulation to be remotely executed using an external simulator developed by the user (optional component). The communication
Figure 1: Petri nets and computational steering framework. The framework consists of four components: steering server, steering graphical user interface (GUI), steering application programming interface (Steering API), and simulators (internal and external). The flow of information goes in two opposite directions: from the simulator to the user (monitoring) and from the user to the simulator (steering). The data structure inside the server is organised in terms of Petri nets: places, transitions, arcs and parameters. The place data structure includes the initial marking, and the transition data structure the rate functions. A model can contain different result views, which are defined by the users and submitted to the server for further download and manipulation.
between these external simulation modules and the other architecture components takes place through the steering APIs. This means that with modest effort, users can include their own favourite simulators and perform the monitoring and steering tasks by help of the other framework components.

2.2 Application Scenario

In a typical application scenario, a user constructs the biochemical reaction network using a Petri net editing tool (e.g., Snoopy). Afterwards, the (stochastic, continuous, hybrid) Petri net model is submitted to one of the running servers to quantitatively simulate it. Later, other users can connect to this model by their steering GUIs. One of the connected users initialises the simulation while other users could stop, pause, or restart it. When the simulator initially starts, it uses the current model settings to run the simulation. Later, other users can remotely join the simulation and change model parameters or the current marking. Figure 2 illustrates graphically a typical application scenario of Snoopy’s computational steering framework.
### 3 System Requirements

In this section we outline the hardware and software requirements to run Snoopy and the Snoopy steering Server (SSServer) on your computer.

#### 3.1 Hardware Requirements

The hardware requirements depend on your specific needs. For instance, if you plan to run big models (100,000 to 1000,000 variables), then higher requirements are needed than to run relatively small ones. At the time of writing this manual, a 64Bit machine, preferably running native Linux is recommended for computational expensive models. Table 1 outlines the minimum and the optimal hardware required to run the Snoopy steering framework.

#### 3.2 Software Requirements

The Snoopy steering framework implementation is platform-independent. Therefore you can use it on your favourite operating system. In more specific terms, you will need one of the following operating systems running on your computer:

- Window XP or higher
- Mac OS X 10.5 or higher
- Linux, e.g., Ubuntu

---

Table 1: Minimal and Optimal Hardware Requirements.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Snoopy</th>
<th></th>
<th>SSServer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimal</td>
<td>Optimal</td>
<td>Minimal</td>
<td>Optimal</td>
</tr>
<tr>
<td>Processor</td>
<td>1 GHz</td>
<td>2 GHz</td>
<td>2 GHz</td>
<td>2.5 GHz or higher</td>
</tr>
<tr>
<td>RAM</td>
<td>256 M</td>
<td>1 GB</td>
<td>512 M</td>
<td>≥ 8 GB</td>
</tr>
<tr>
<td>Free Hard Disk Space</td>
<td>500 M</td>
<td>2 GB</td>
<td>500 M</td>
<td>10 GB</td>
</tr>
<tr>
<td>LAN adapter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

*This table is both for 32 and 64 Bit machines.*
Next, you will need a copy of Snoopy suitable for your operating system which can be downloaded from http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy.

Finally, a copy of the SSServer is required to run this framework. Please note, at the time of writing this manual, the SSServer is not available at Snoopy’s website. However, it can be requested from the authors by email.

4 Installation

The specific installation procedure depend on your operating system version. Below we give two examples using Windows and Mac OS X.

4.1 Install Snoopy under Mac OS X

To install Snoopy under Mac OS X, first acquire a Snoopy version. The Snoopy setup file under Mac OS is in a disk image format (dmg). All the necessary data are provided in a single file. To install this file on your computer do the following steps:

- Mount the disk image on your computer by double-clicking the dmg file.
- The disk image will appear as another CD in your Finder with the name ”Snoopy”.
- Opening this disk, you will find the Snoopy application bundle, as shown in Figure 3.
- Copy Snoopy to your desired location, or drag and drop it into the application folder.
4.2 Install SSServer under Mac OS X

To install the SSServer to your Mac OSX, similar steps as for installing Snoopy are required. We repeat them again for your convenience.

- Mount the disk image on your computer by double-clicking the dmg file.
- The disk image will appear as another CD in your Finder with the name ”SSServer”.
- Opening this disk, you will find the SSServer application bundle, as shown in Figure 4.
- Copy the SSServer to your desired location, or drag and drop it into the application folder.

4.3 Install Snoopy under Windows

To install Snoopy under Windows, follow these steps:

- Obtain the windows installer package for Snoopy. This should be a msi file.
- Double click the obtained setup package.
- The windows installer will start as shown in Figure 5.
- Follow the simple instructions in this window by hitting the next button.
- At the end of these steps Snoopy should be installed on your computer and a shortcut will be created at your desktop.
- To start Snoopy double click the Snoopy’s shortcut from your desktop.
4.4 Install SSServer under Windows

Similar steps are required to install the SSServer under Windows. For your convenience we repeat them in the following procedure.

- Obtain the windows installer package for SSServer. This should be a msi file.
- Double click the obtained setup package.
- The windows installer will start as shown in Figure 6.
- Follow the simple instructions in this window by hitting the next button.
- At the end of these steps SSServer should be installed on your computer and a shortcut will be created at your desktop.
- To start the SSServer, double click the SSServer’s shortcut from your desktop.

5 Getting Started

5.1 Launching Snoopy and SSServer

The first step to use Snoopy in the steering mode is to open the Snoopy Steering Server (SSServer) on your local computer or on another remote machine where you want to run the simulation. Second you need to open Snoopy on your machine. The exact procedure
depends on the operating system you use. We assume that you have basic knowledge to run software on your operating system.

After starting the server, it will look as shown in Figure 7 under the Mac OS X. Other platforms might have slightly different, but similar interfaces. There are two important information you need to notice after starting the SSServer: the running server name (IP) and the listening service number. They are written in the log window after the SSServer started successfully. You will need these information when connecting to a server using Snoopy. For instance, in Figure 7, the server name is swqlab3.informatik.tu-cottbus.de and the IP is: 141.43.202.57. The server is listening to the service number 3000.

In the status bar, there are some useful information about the status of the SSServer. The first one gives information about the current state of the server. For example in Figure 7 the SSServer state is "Running". Other possible states are "stopped" and "Restarted". Make sure before connecting to a server that its state is "Running". The second information in the status bar is the number of currently running models inside the server. Each time you submit a new model, this number will increase. The third and final information is the number of clients (users) that are currently connected to the server.

After you have started the SSServer, you need to open Snoopy on your machine. Figure 8 gives a screenshot of Snoopy under Mac OS X.
Figure 7: Snoopy Steering Server under Mac OS X

Figure 8: Snoopy under Mac OS X
5.2 Connecting to the Server

To connect to a running SS Server using Snoopy, you first need to decide if you would like to use a model from those running on the SS Server (if there are some) or if you would like to submit a new model. In the latter case you will need to open your Petri net model using Snoopy. Nevertheless, to open the connection window, do the following steps in either of these cases:

1. From the view window, select ”Start Steering-Mode” (see Figure 9) or just press ”F7” from the function keys if you prefer using short cut keys.

2. The connection window will be opened (see Figure 10).

After you have opened the connection window you can connect to the sever using the default setting via the ”connect” button. This makes sense if you run the SS Server at the same computer as you use Snoopy or if you already configured Snoopy to connect to a certain SS Server. However, if you run the SS Server on another computer, you must first configure the connection before pressing the ”connect” button, see Section 5.1.

Another important point is that the SS Server and Snoopy should have the same version number of the communication library. You cannot let Snoopy connect with an SS Server having a different version. A check will be done at the beginning and you will be informed if an incompatible version is detected. In this case the Snoopy and SS Server versions are displayed for your convenience.
5.2.1 Configuring Connection Settings

To configure the connection settings to connect to a certain SSServer do the following steps:

- Press the "Show details" button from the connection window.
- New fields appear as illustrated in Figure 11.
- In the service text box, enter the service number as discussed in Section 5.1.
- In the server text box, enter the server name or the IP address.

The default value of the service number is "3000", and the default for the server name is "localhost". To simplify entering these information, you can also select from previous setting using a list.

5.2.2 Saving Connection Settings

The connection settings, which you have entered in the previous step, are automatically saved to the Snoopy registry. Therefore you do not need to re-enter them again when you later want to re-connect to the same SSServer.

5.3 Submit a New Model

To use a new model in the steering mode do the following steps:

1. Open the Petri net file in Snoopy.
2. Open the connection window as discussed in Section 5.2.

3. Make sure that you correctly configured the connection settings to connect to a running server.

4. Hit the "Connect" button. The steering option window will appear as illustrated in Figure 12.

5. Make sure, the "send a new model" option is selected.

6. Enter a model name and then press ok.

   Please note that the name of the loaded file in Snoopy is used as a default name for the new model. You can edit the name as you want under the condition that there is no model running on the SSServer side with the same name.

   After pressing the Ok button, the steering dialogue will open as shown in Figure 13.

   Please note, if you use a coloured model, then the unfolding window will automatically appear after step 6, because you need to unfold the model before submitting it to the server. To unfold a model, press the "Start" button on the unfolding window as shown in Figure 14.

5.4 Using an Existing Model

The Snoopy steering framework gives you the chance to use existing models that are running on the SSServer without opening them in Snoopy. For example, let us assume that you have one model that is already running in the SSServer and you want to disconnect and return back to it later. In this case, you do not need to reopen the Snoopy file or even to unfold it if it is a colored model. To steer an existing model, follow this procedure:

1. Open Snoopy, and then open the connection window as described in Section 5.2.
2. Press the "Connect" button.

3. A new window will appear with the models currently running on the SSServer.

4. Select one of these models and press ok.

5. The steering dialogue will open with the selected model loaded.

Please note, if there are no running models on the server side, then you will be informed and the process of opening the steering GUI will terminate. Besides, you can do these steps while a Petri net model is opened in Snoopy. However, you will need to select the option "Use an Existing Model". After that you can select one of the running models from the list.
5.5 Replacing an Existing Model

When constructing a Petri model for the first time, you might need to submit the same model several times with some modifications. Such modifications might involve adding new places or new transitions to the model definition. In this case, submitting the same model with a different name will increase the number of models running on the SSServer. A better procedure is to submit the modified model with the same name as the old one. In this case the modified model will replace the previous one. Please note, in order to replace an existing model, there should be no other users connected to it. If there are other users connected, the server will refuse to replace this model, and the client will get a notice.

6 Further Reading

