MULTI-AGENT APPROACH TO TRAFFIC SIMULATION IN NETLOGO ENVIRONMENT – LEVEL CROSSING MODEL

Aleš Janota – Karol Rástočný – Jiří Zahradník

{ ales.janota | karol.rastocny | jiri.zahradnik }@fel.utc.sk
Structure of the presentation

1. Introduction

2. Brief introduction to the NetLogo environment
   (what is it?, availability, main screens, menus and control/indication elements, ....)

3. Description of „Level Crossing“ Model
   (modelled functions, used simplifications, ....)

4. Practical demonstration of the model running

5. Conclusions
What is NetLogo?

• A programmable multi-agent modelling environment for simulating natural and social phenomena
  decentralized, interconnected nature of phenomena

• Written entirely in Java (version 1.4.1)

• Current version (October 2005): 3.0

• Runs on all major platforms (MacOS, Windows, Linux, et al), as a standalone application; individual models can be run as Java applets inside a web browser.
Availability

• NetLogo has been under development since 1999

• Created by Northwestern University's Center for Connected Learning and Computer-Based Modeling

• Freeware: http://ccl.northwestern.edu/netlogo/

• Comes packaged with extensive documentation and tutorials and a large collection of sample models (a library with over 150 sample models) for easy access
Main window: Menu + 3 Tabs

WHAT IS IT?

This section could give a general understanding of what the model is trying to show or explain.

HOW IT WORKS

This section could explain what rules the agents use to create the overall behavior of the model.
NetLogo’s World: 2D (3D)

Example of a turtle (car)

A grid of patches

Coordinates

(pxcor, pycor)

(xcor, ycor)

TST ’05 – Poland / 3-5 November 2005
“Interface” Tab - Structure

By default:
\[ \text{screen-edge-x} = \text{screen-edge-y} = 17, \]
\[ \text{i.e. } 35 \times 35 = 1225 \text{ patches total} \]

- Interface toolbar
- Setting size
- Simulation speed
- Turtles shape
- Freezing display

For giving immediate commands
“Interface” Tab – Toolbar

Control items

Indication items
Recommended convention

1. What is it? ... what the model is trying to show or explain
2. How it works? ... what rules the agents use to create the overall behaviour of the model
3. How to use it ... how to use the model, including a description of items in the interface tab
4. Things to notice ... some ideas for the user to notice while running the model
5. Things to try ... what could the user try to do (move sliders, switches etc.) with the model
6. Extending the model ... some ideas to add or change in the procedures tab to change the model
7. NetLogo features ... things worth of user's interest, unusual features of the model etc.
8. Related models ... the names of models which are of related interest
9. Credits and references ... e.g. reference to the model's URL on the web
Procedures Tab

A working area where source code of the model is placed.

Commands saying what agents should do

Reporters performing operations and informing commands or another reporters about results

Basic built-in functions

Primitives

Procedures

User defined functions
Known traffic simulation models

   http://ccl.northwestern.edu/netlogo/models/TrafficBasic. Center for Connected Learning
   and Computer-Based Modeling, Northwestern University, Evanston, IL, 1997.
   http://ccl.northwestern.edu/netlogo/models/Traffic2Lanes. Center for Connected
   Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1998.
   http://ccl.northwestern.edu/netlogo/models/Gridlock. Center for Connected Learning and
   Computer-Based Modeling, Northwestern University, Evanston, IL, 2002.
    http://ccl.northwestern.edu/netlogo/models/HubNetGridlock. Center for Connected
    Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 2002.
     Intersection Control Mechanism. In Proceedings of the Third International Joint
     Conference on Autonomous Agents and MultiAgent Systems, p. 530–537, ACM. 2004
Modelled features

**Controlled features:**
- Setting a time for generating new vehicles
- Setting max. velocity of road vehicles
- Setting velocity of the train
- Setting a number of train units (up to 3)
- Installing a traffic sign for speed limitation (20 km/h), for each direction separately
- Setting a variable time period to closing barriers (the value depends on the current speed of the train approaching the level crossing - simplified to two stages <=/> 40 km/h)

**Monitored features:**
- Total cars passed (for each direction and totally)
- Car throughput (for each direction and totally)
- Current number of cars on the screen
- Plot „Throughput vs time“; elapsed time
Train running through the level crossing

Approaching section

Detection section

Annulation section
Turtle shapes
Conclusions

- Motivation for writing and presenting this paper
- Existing traffic models mapped and surveyed
- The level crossing model created without pretension to model exactly and strictly all functions of the level crossing installation (i.e. with many simplifications!!! warning lights displayed in a non-standard way, train is running only in one direction, the barriers will not open if train is not coming for a longer time, there is no emergency control of barriers here etc. etc)
- Many other NetLogo features being worthy of mention (HubNet simulations based on a client-server architecture, System Dynamics modeller, 3D modelling, ...)

TST ’05 – Poland / 3-5 November 2005
Thank you for your attention ..... 

This work has been partially supported by the Grant Agency of the Slovak Republic VEGA, grant No. 1/1044/04 “Theoretical Foundations for Implementing e-Safety Principles into Intelligent Transportation Systems” and partially by the institutional research project No. 07/604/2005 “Transport telematics and tools to improve its quality” solved at the Faculty of Electrical Engineering, University of Žilina.