SLX-MODEL AND PROOF ANIMATION™ BASED VISUALIZATION OF THE RIGA BALTIC CONTAINER TERMINAL

Steffen Straßburger, Thomas Schulze
Department of Computer Science
Otto-von-Guericke University
Universitätsplatz 2
30106 Magdeburg, Germany
E-mail: {strassbu, tom}@isg.cs.uni-magdeburg.de

Juri Tolujev
Riga Technical University
Department of Modeling and Simulation
Riga, LV 1658, LATVIA
E-mail: tolujev@itl.rtu.lv

KEYWORDS

Simulation, SLX, Container terminal.

ABSTRACT

The paper describes the SLX-based simulation of the Riga Baltic Container Terminal (BCT) and the visualization of the model based on Proof Animation for Windows. The work is a result of the INCO2 1997 Project PL976012 DAMAC HP, funded by the European Commission, DGXIII.

SLX-BASED SIMULATION

SLX is a discrete event simulation tool for the Windows 95/98/NT/2000 operating systems developed by the Wolverine Software Corporation (Henriksen 1997a). SLX is a classical simulation language-oriented stand-alone tool that includes a programming language with a C-like syntax. SLX is the designated successor of the well-known GPSS/H.

When transferring an existing GPSS/H simulation model to SLX, the user has two general options for doing this:

a) The user can use the SLX-hosted implementation of GPSS, contained in the H5/H6.SLX modules. This implementation was built to support the adoption process of old GPSS users converting to SLX. At the same time it demonstrates one of the main strengths of SLX, its extensibility. Within the SLX language the users can define their own statements and thus build simulation packages on top of the actual SLX language. This has been done with the GPSS language. The SLX-hosted implementation of GPSS supports the GPSS language to a large extend, but not completely. Some re-writing of GPSS models may be necessary to adopt them to make them run under SLX.

b) The other choice is to completely re-develop an existing GPSS/H model with the native SLX constructs and approaches. While at the first view this seems to be the more time-consuming approach it also has several advantages. In a lot of cases, techniques used for producing good GPSS/H models result in bad SLX models, if models are transliterated from GPSS/H to SLX, or if new SLX models are developed imitating GPSS/H style. Probably the biggest offending technique is using integer indices to access entities. In GPSS/H, this is the only way to access a facility in the single collection of facilities available at run-time. In SLX, one can have arrays of facilities, facilities as sub-objects, dynamically created facilities, etc. Failure to take advantage of these improved ways of organizing data and objects is resulting in badly readable SLX models.

IMPLEMENTATION OF THE SIMULATION MODEL FOR RIGA BCT

In the development of an SLX based simulation model of the Baltic Container Terminal (BCT) it also had to be decided which strategy was to be used. An GPSS/H model of the harbor was existing (at least in major parts). In first experiments it was tried to use approach a) described above. Soon the disadvantages of this approach were discovered and the development efforts were redirected to use approach b). One extra-advantage of this approach is that native SLX models can easily fit into the HLA world view of objects and interactions (Straßburger 1998, 1999). This tremendously facilitates the interoperability aspects, e.g., if in future the BCT model should be coupled with other components or programs.

The implementation of the BCT in SLX has been structured in the following way: For all moving entities of the model, active SLX object classes have been defined. SLX distinguishes between active and passive objects. While passive objects merely function as complex data types, i.e., a collection of attributes without own functionality, active objects also define the properties, i.e., the behavior, of a class.

The active objects contained in the model include the following classes:

- cl_Ship models the arriving ships, their unloading and loading processes, and their departure
- cl_Trailer models the trailers moving through the container yard
- cl_Crane models the different cranes located in the harbor. Different sub-types exist for instance for the quay cranes and the railway cranes
- cl_Train models arriving and departing trains and their loading and unloading
- cl_Truck models truck sets which arrive and deliver/receive containers

Fig. 1 and 2 depict the relationships between the most important classes of the model and also shows their attributes and properties.

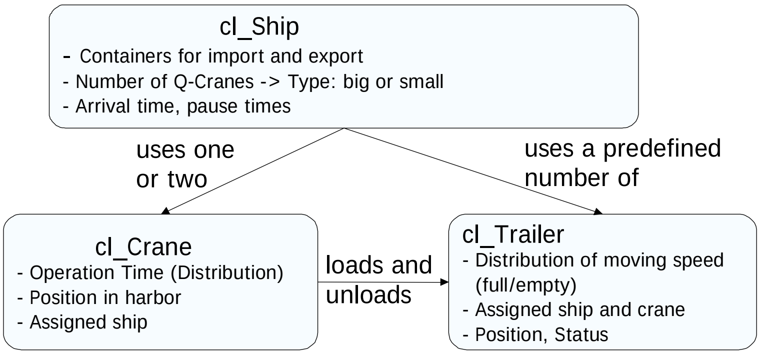

Figure 1: Relationship of Important Classes of the SLX Model and their attributes

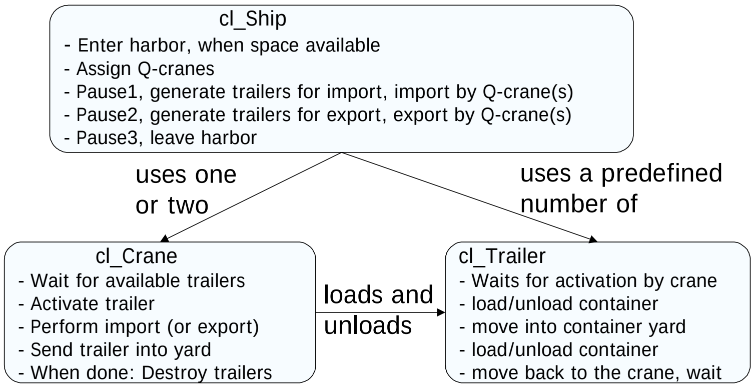

Figure 2: Relationship of Important Classes of the SLX Model and their behavior

The management of resources in the SLX model is performed using sets. Sets are a common way of modeling the management of resources in the SLX language. In the model, all quay cranes are, for instance, part of the set "AllCranes" which marks its members as available. When cranes are assigned to a ship they are removed from this set and placed into the ship's set "MyCranes". Thus they are marked as currently busy serving a certain ship. A similar strategy is used for trailers. Trailers are initially created for a certain ship, placed into the ship's set of available trailers, and removed by a crane once the crane is ready to load/unload the trailer.

VISUALIZATION OPTIONS BASED ON PROOF ANIMATION

Proof Animation is a general-purpose animator designed for use with the widest possible variety of simulation tools. Every Proof animation requires two ASCII input streams,

- a layout stream, describing static characteristics of an animation, e.g., the background drawing over which objects move, and
- a trace stream, which is a time-ordered sequence of commands which create, destroy, move, and otherwise change objects displayed on the layout, portraying events in a simulation.

Both of these streams are free-format ASCII text, with well documented ("open") architectures which can be generated easily in a variety of ways (Henriksen 1998, 1999).

Proof can be used in post-processing mode or directly driven by another program. When Proof is used in post-processing mode, its input streams must be stored in files. Trace files are almost always written using a simulation language or package, such as SLX (Henriksen 1997b), GPSS/H, Extend, Slam, Siman, Simscript II.5, etc. Layout files are almost always developed using Proof's built-in drawing tools. Proof also includes a CAD import feature, allowing importing of .DXF files.

When Proof is directly coupled to simulation software, input streams are transmitted to Proof one line at a time via subroutine calls. Proof can be directly driven by any program which is capable of constructing C-compatible Dynamic Link Library (DLL) calls; i.e., the directly driven version of Proof is packaged as a Windows DLL.

IMPLEMENTATION OF THE VISUALIZATION OF THE BCT MODEL

The simulator SLX is tightly coupled with the DLL version of Proof and could thus be easily used for performing on-line visualization of the BCT model. Since the target scenario which was to be achieved within the project has SLX run in a distributed environment on a web-server, the traditional post-processed animation option was chosen as the default visualization. Optionally it is possible to switch to a local on-line animation, if SLX and Proof are available locally. This can be done easily with minor modifications of the model, i.e., by exchanging the default include file "proof3.slx" (for post-processed animation) to "p4dll.slx" (for on-line visualization).

The implementation of the visualization of the SLX-BCT model makes extensive use of the path concept of Proof. All important points in the layout, e.g., position of cranes, loading and unloading stations for ships and trains, positions of container yards, etc. are assigned a unique number in the layout. Between each of the points paths following a certain naming convention exist. A path connecting position 17 (the left most quay crane in Fig. 3) and a position 3 (one of the container yards) would be named "pa1703". In the opposite direction, a path named "pa0317" exists. All movements of vehicles, e.g., trailers and trucks, take place on these paths.

The SLX model dynamically constructs the paths which vehicles like trailers use. It analyzes the input files for the distances between certain points and adjusts the speed of trailers accordingly. Fig. 3 shows an example of the path layout of the visualization layout.

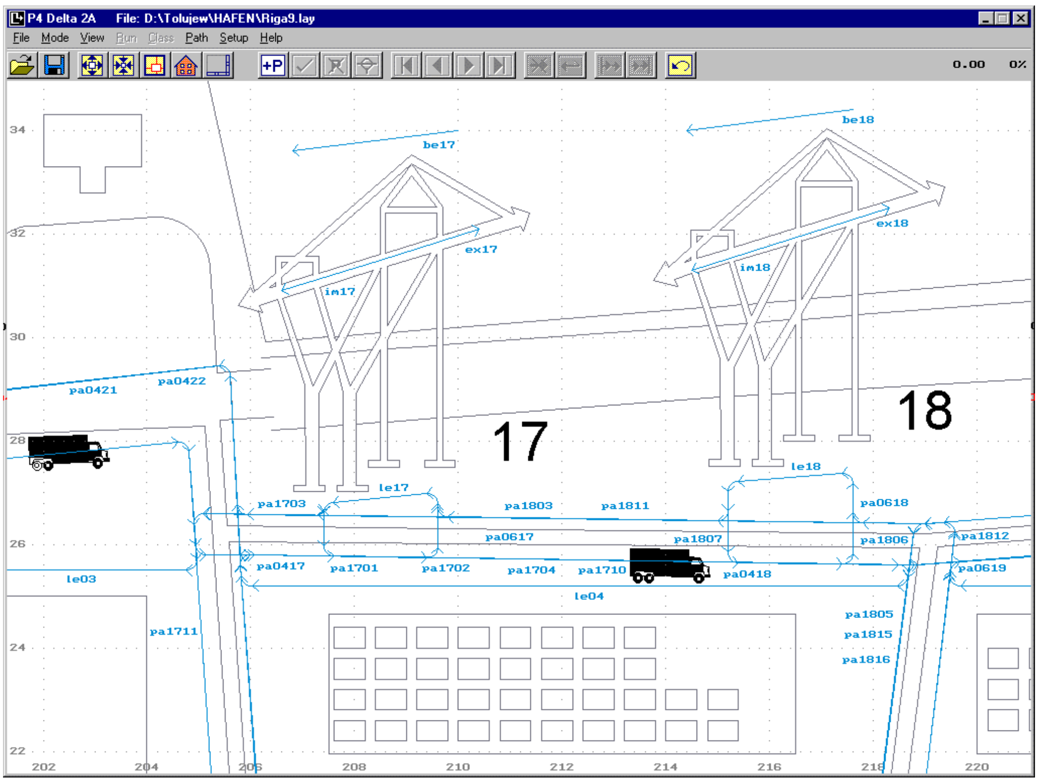

Figure 3: Screenshot showing the path layout of the Proof Animation of the BCT

SUMMARY

The combination of SLX and Proof Animation provides powerful mechanisms for building models in a high performance simulation language and performing visualization in post-run as well as on-line mode.
The combination of both tools has special advantages when it comes to dynamically changing the layout. The SLX model is completely independent from the animation model in terms of layout issues as long as the path naming conventions are kept. This way it is easily possible to experiment with different routing strategies in the container yard. All relevant data that the SLX model needs to know is provided by the input data file. This file can be created in different ways: manually, by an automatic analysis of a Proof layout file, or by assuming certain standard values.

REFERENCES

Henriksen, J. O. 1997a. An Introduction to SLX. In Proceedings of the 1997 Winter Simulation Conference, eds. S. Andradóttir, K.J. Healy, D.H. Withers, B.L. Nelson, pp. 559-566.

Henriksen, J. O. 1997b. SLX and Proof Animation. In Deussen, O. and P. Lorenz (Ed.), Proceedings of the Simulation and Animation Conference Magdeburg, March 6.-7., 1997. SCS European Publishing House San Diego / Erlangen / Ghent / Budapest 1997, pp. 287-294.

Henriksen, J.O. 1998. Windows-Based Animation with Proof™. In Proceedings of the 1998 Winter Simulation Conference, eds. Medeiros, D., E. Watson, J. Carson, and M. Manivannan, pp. 241-247. SCS, Washington

Henriksen, J.O. 1999. General-Purpose Concurrent and Post-Processed Animation with Proof™, In Proceedings of the 1999 Winter Simulation Conference, eds. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, pp. 176-181. SCS, Phoenix.

Straßburger, S. On the HLA-based Coupling of Simulation Tools. In Proceedings of the 1999 European Simulation Multiconference, ed. H. Szczerbicka, pp. 45-51 (Vol. 1). SCS, Warsaw, Poland.

Straßburger, S., T. Schulze, U. Klein, J. O. Henriksen. 1998. Internet-based Simulation using off the shelf simulation tools and HLA. In Proceedings of the 1998 Winter Simulation Conference, eds. Medeiros, D.J. and E. Watson, Washington D.C.

ABOUT THE AUTHORS

STEFFEN STRASSBURGER holds a Master's degree in Computer Science from the Otto-von-Guericke University in Magdeburg, Germany. He is currently working towards his PhD at the Institute for Simulation and Graphics at the same university. His experience with inter-networking and simulation includes a one-year stay at the University of Wisconsin, Stevens Point and a stay at the Georgia Institute of Technology, Atlanta. His main research interests lie in distributed simulation and the High Level Architecture.

THOMAS SCHULZE is an Associate Professor in the School of Computer Science at the Otto-von-Guericke-University in Magdeburg. His research interests include modeling methodology, public systems modeling, manufacturing simulation, and distributed simulation with HLA. He is an active member in the ASIM, the German organization of simulation.

JURI TOLUJEV is an Associate Professor in the Department of Modelling and Simulation at Riga Technical University. His research interests include simulation modeling and analysis, simulation models building, verification, and validation based on record data interpretation.