Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 155 (2019) 654–659

Procedia Computer Science

www.elsevier.com/locate/procedia

The 6th International Symposium on Emerging Inter-networks, Communication and Mobility (EICM)

August 19-21, 2019, Halifax, Canada

OMNET++ and Maple software environments for IT Bachelor studies

Ivanna Dronyuk[a], Olga Fedevych[a], Roksolyana Stolyarchuk[a], Winfried Auzinger[b]*

[a] Lviv Polytechnic National University, Bandery str.12, 79013 Lviv, Ukraine

[b] TU Wien, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

Abstract

This article deals with the methodology of computer network modeling for scientific and educational purposes. We choose tools which are useful, easy to understand and enable to model real life communication tasks for IT Bachelor students.

Keywords: simulation; computer networks; software tools; computer networks modeling, IT Bachelor degree.

1. The problem of creativity in IT bachelor studies

The purpose of this article is to investigate the most suitable software environments which student could use in order to understand different theoretical aspects of computer network modelling. The described software can be used to teach different mathematical disciplines, for example discrete mathematics, computer algebra, computer linguistics and others, that are necessary for the formation of a high level of competence of students of specialties "Computer Science" and "Information Systems and Technologies" for bachelor and master's level of training. Students' simulation of the computer network of the department, university, institute, and on this basis, a real improvement of the network's work awakens the students' genuine interest in scientific research. Also, the use of

* Ivanna Dronyuk. Tel.:+38-050-430-3378; fax: +38-032-258-2680.
E-mail address: ivanna.m.droniuk@lpnu.ua

1877-0509
10.1016/j.procs.2019.08.093

high tech level software promotes an active engagement of students in research. This article discusses the application of Maple and OMNeT++ software environments, and also a methodology for teaching disciplines related to computer networks and computer algebra making use of these packages for the training of students of modern information technologies is discussed.

The network simulator OMNeT++ can be recommended for in-depth study of individual sections of the course "Computer networks" in the course design and implementation of master works. The OMNeT++ environment using the INET framework makes it possible to explore network models in a step-by-step mode, as well as to obtain generalized statistical characteristics. The INET framework components ready to describe the various network elements provides ease of implementation for rather complex network models.

2. Computer Network Modeling and Analysis in Maple

In Computer Algebra (CA) systems like Maple [9] there are a number of special packages e.g. for supporting discrete mathematics. In particular, in Maple the GraphTheory package (replacing an older package called networks) serves for creating, manipulating and analyzing graphs, and for importing graphs in various standard formats.

In Maple, a graph is represented by a special internal data structure; undirected or directed as well as unweighted or weighted types are supported and can be generated using the appropriate constructors.

In our context, a network is a directed graph containing at least one source and at least on sink. Several standard algorithms are readily implemented, e.g. for finding a shortest path or for solving a max-flow problem.

There is an ongoing discussion about the question how to make use of CA system for educational purposes. In the context of graph theory and network problems, there are several advantages:

- Graphs can be generated using CA, and their structure can be manipulated in an easy way.
- CA supports visualization in an automatic way, which is very useful and saves hand work (see Fig. 1).

Further aspects are related to programming and modeling:

- If you want to set up and manipulate your individual nontrivial graph, you cannot do it by just passing the data into the generator function in a straightforward manner, but you have to understand the underlying data structure and how to use programming features like loops to realize this.
- Once you have understood this, you can proceed with the modeling aspect, that is, translating a given problem into the graph-theoretic context, setting up the graph in CA and solve a given problem, e.g., by available standard algorithms or by your own solution algorithm.

Concerning the modeling aspect, this is a didactical issue not directly related to the use of CA (however, from CA some benefit on the experimental side may be useful). Anyway, as an illustration we now present a simple, very well-known classical example, where a problem concerning a feasible flow through a network is appropriately modeled and solved by CA.

Example: A man is wandering around, and with him he has a wolf, a goat, and a cabbage. He has to cross a river by boat, but the boat is so small that he only can cross either alone or take either the wolf, the goat, or the cabbage with him for a crossing. Now the question is: How to organize this, using an even number of crossings between riversides 1 and 2, such that everything will stay undamaged:

- It is not possible to leave the wolf together with the goat, because the wolf will eat the goat.
- It is not possible to leave the goat together with cabbage, because the goat will eat the cabbage.
- On the other hand, the wolf is not interested in eating the cabbage.

How to proceed? For some basic graph-theoretical background see for instance [1]. Consider all possible states in course of a multiple forward/backward river crossing. These are given by the possible positions of the man (M), the wolf (W), the goat (G), and the cabbage (C) either on riverside 1 or on riverside 2. This is equivalent to the information about which subset of {M,W,G,C} is on riverside 1. The number of a priori possible constellations is 2^16, the number of subsets of {M,W,G,C}.

Now we consider *admitted constellations* only, i.e., those which are compatible with the given side conditions. This means that {W,G,C}, {W,G}, and {G,C} are not admitted (this refers to riverside 1); furthermore, {M}, {M, C} and {M,W} are not admitted (this refers to riverside 2). In this way we end up with a set of 10 admitted constellations (i.e., admitted subsets of {M,W,G,C}), which we now consider as the nodes of a directed unweighted graph. Actually, this graph is a network with source {M,W,G,C} and sink {}. Our next job is to define the edges in this graph, i.e., the possible movements, which is a routine job. For instance, {M,W,G,C}→{W,C} and {M,G,C}→{G} are edges of this graph, i.e., they represent admitted crossings.

As soon as this is accomplished, we can generate the graph in our CA system and use a standard algorithm for finding the (shortest) path from source to sink. Here the 10 admitted states are labeled by numbers 1-10 in the following way,

1	2	3	4	5	6	7	8	9	10
{C,G,M,W}	{G,M,W}	{C,M,W}	{C,G,M}	{G,M}	{C,W}	{W}	{C}	{G}	{}

For instance, ‘1’={C,G,M,W} represents the initial state (all on riverside 1), and ‘10’={} represents the end state (all on riverside 2). See Fig.1, where two solutions of the river crossing problem are visualized.

Below we also display a fragment of our Maple code for this problem (see Fig. 2).

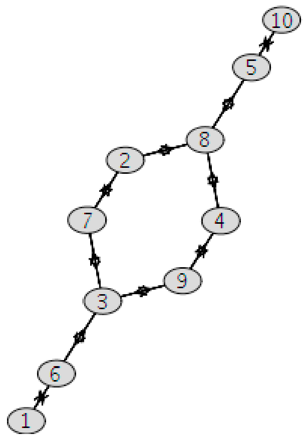

Fig. 1. Graph representation in Maple.

One may ask whether this CA-based approach makes sense, since this problem is well-known and popular quiz which can be solved by hand without software support. The answer is positive: It serves a didactical purpose, and as more complex network problems are to be considered, the proper modeling and appropriate use of software is mandatory.

3. Network simulation using OMNET++

The software package OMNeT++ is a high-quality tool for simulating the performance of computer networks. It can be used while preparing solutions for various problem in different domains, as: modeling of wired and wireless computer networks, various protocol modeling, modeling of queues, modeling of multiprocessors, modeling of other distributed systems, evaluating performance of complex software solutions. Despite that, this environment allows modeling and simulation of any software, where the discrete event approach is used, and can be conveniently mapped into entities that are communicating step by step by using message exchange [2].

OMNeT++ has the resources to build computer networks of different sizes and different topology architectures [3]. It contains a large database of ready-to-use network elements that already include simulation algorithms and computer information processing steps, including time, transaction costs.

```
> restart:
> with(GraphTheory):
  with(combinat):
> #
> # MWGC problem
> #
> # Define possible states (a priori, i.e., independent of side conditions):
> #
> n:=4:
> states:=convert(powerset({M,W,G,C}),list):
> #
> # Define nodes of graph by excluding states not admitted:
> #
> nodes:=[]:
  for state in states do
      # forbidden at riverside 2:
      if evalb(member(W,state) and member(G,state)
               and not member(M,state)) then next end:
      if evalb(member(G,state) and member(C,state)
               and not member(M,state)) then next end:
      # forbidden at riverside 1:
      state:={M,W,G,C} minus state:
      if evalb(member(W,state) and member(G,state)
               and not member(M,state)) then next end:
      if evalb(member(G,state) and member(C,state)
               and not member(M,state)) then next end:
      # admitted:
      nodes:=[seq(nodes),state]:
  end:
> # ==> admitted states:
  numelems(nodes),nodes
        10, [{C, G, M, W}, {G, M, W}, {C, M, W}, {C, G, M}, {G, M}, {C, W}, {W}, {G}, {C}, ∅]
> inodes:=table([seq(j=nodes[j],j=1..numelems(nodes))]):
> inodes_inv:=table([seq(nodes[j]=j,j=1..numelems(nodes))]):
> #
> # Define edges of directed graph by excluding moves not admitted:
> #
> edges:={}:
> for node1 in nodes do
      for node2 in nodes do
          # no loops:
          if node1=node2 then next end:
          # permitted: M must cross, W,G, or C may also cross with him:
          M_crosses:=evalb(member(M,node1) xor member(M,node2)):
          W_crosses:=evalb(member(W,node1) xor member(W,node2)):
          G_crosses:=evalb(member(G,node1) xor member(G,node2)):
          C_crosses:=evalb(member(C,node1) xor member(C,node2)):
          admitted:=M_crosses and
                     ((not W_crosses and not G_crosses and not C_crosses) or
                      (    W_crosses and not G_crosses and not C_crosses) or
                      (not W_crosses and     G_crosses and not C_crosses) or
                      (not W_crosses and not G_crosses and     C_crosses)):
          if admitted then
             edges:=edges union {[node1,node2]}:
          end:
      end:
  end:
```

Fig. 2. Fragment of Maple code (setup of graph).

It is known [4] that the results of theoretical approximations of mathematical models are tested on the basis of the OMNeT++ package.

OMNeT++ itself provides infrastructure as a component architecture for simulation models. Models are assembled from reusable components also know as termed modules. Such written modules could be used in other models within other circumstances, and can be combined in various ways like bricks.

OMNeT++ projects can be realized under various UI's. Especially useful for students of different courses may be graphical and animating user interfaces, that can be used by them for demonstration and debugging purposes. Among them, command-line user interfaces can be used for batch execution practices.

OMNeT++ is highly portable. It was tested on the most common operating systems, for example: different versions of OS Windows, Linux, and Mac OS/X. This environment could be compiled out of the box or after small modifications on other Unix-like OS's.

This software also provides parallel-distributed simulation, and can use some mechanisms for communication between parts of a parallel-distributed simulation, for example named pipes. The parallel simulation process also can be extended, or new algorithm can be plugged in. A big advantage in is the fact that created models do not need any additional special instrumentation to be run in parallel mode – this can be manually configured.

The OMNeT++ environment can also be used for student's presentation of parallel simulation processes, because previously described simulations can be run like parallel processes even under using the GUI that provides detailed feedback of work. In this case, the teacher can suspend the simulation process, check and estimate the required information, make clarifications to the parameters of the simulation process and continue or stop simulation if needed.

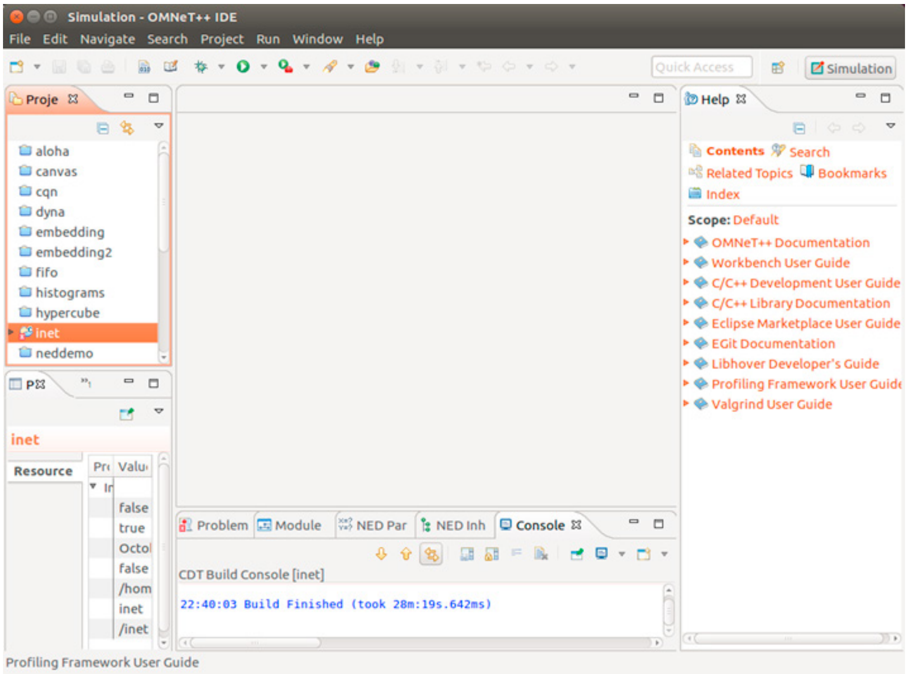

Fig.3 OMNeT++ interactive interface.

Nevertheless, OMNeT++ allows to group simple modules into compound modules and so forth; the number of such hierarchy levels cannot be limited (see Fig. 3). Created for such purposes, the OMNeT++ Integrated Development Environment is based on the Eclipse platform, and extends this platform with brand new views, wizards, editors and specific functionality. This software adds possibilities for creating, preparing and configuring different models (NED or ini files), execute batch processes, and will provide an opportunity to analyze simulation results, while the Eclipse environment provides C++ programming language editing, integration with various version control systems. Among them, one can make use of other optional features (bug tracker, UML modeling, data base access) via various plug-ins.

In the OMNeT++ environment one can simulate and explore various protocols for wired and wireless networks: IPv4, IPv6, TCP, UDP, 802.11, Ethernet, OSPF, MPLS with LDP, RSVP-TE, and many others.

In recent years, the concept of Internet of Things (IoT) has become increasingly important, and therefore there is an increasing need for the study and research of wireless sensor networks. For their simulation in the OMNeT++ environment, the MiXiM framework was developed. MiXiM is an OMNeT++ simulation platform designed specially for mobile and fixed wireless networks, including touch networks such as ZigBee. It offers detailed models of radio wave distribution, possibility to study power consumption, influence of types of routing on the operational characteristics of networks, and provides analysis tools for MAC-protocols and multiple access methods to the physical layer for the prevention of collisions CSMSA/CA. Some model for services are presented in [5].

To study networks using simulation in the OMNeT++ environment, one can recommend a method proposed in [6,7] for sequential complication of network models taking into account features of existing standard protocols and network implementations. Technique for constructing simulation models of networks with a description of the main stages of simulation, methods for launching the network model and analysis of the results are presented in this paper. The method is illustrated by the example of Ethernet network simulation. The examples for application simulation in OMNeT++ are presented in [8]. The similar approaches for stimulation student's studies activities are considered in [10,11].

The commercial version of OMNeT++ is called OMNEST. OMNeT++ itself is free in use for non-commercial models and can be used for teaching in educational institutions.

These options of choosing a simulator can provide an ability for students to master their skills in network architecture, topology and functionality modeling.

4. Conclusion

The software tools described here can be used at universities as for serious scientific research and for in-depth training of students of disciplines related to information technologies. Also, the use of high tech level software promotes an active engagement of students in research. Thus, the described software packages are a good tool for increasing the interest of students in scientific work.

References

[1] B.L. Schwartz: An Analytic Method for "Difficult Crossing" Puzzles, in Mathematics Magazine 34/4,1987-193,1961.

[2] https://omnetpp.org/

[3] Nurminen J.N.: Using software complexity measures to analyze algorithms - an experiment with the shortest-paths algorithms / Nurminen J.N. // In Computers & Operations Research. Elsevier Science. – 2003. – pp 1121-1134.

[4] Nycz T. A Numerical Comparison of Diffusion and Fluid-Flow Approximations Used in Modelling Transient States of TCP/IP Networks / Nycz T., Nycz M., Czachórski T. // 21st International Conference CN 2014. – Brunów, Poland. – 2014. – pp. 213-222.

[5] P. Balco, M. Greguš, N. Kryvinska, "Education as a Service (EaaS) for Organizations", The Spring Servitization Conference 2014 (SSC 2014), 12-14 May 2014, Aston Business School, Aston University, UK, Eds. T. Baines, B. Clegg, D. Harrison, ISBN 978-1-85449-472-6, pp. 174-181.

[6] S. Kaczor, N. Kryvinska, It is all about Services - Fundamentals, Drivers, and Business Models Journal of Service Science Research, Springer, 5 (2) (2013), pp. 125-154

[7] Panova, O., Obelovska, K. An adaptive ACs number adjusting algorithm for IEEE 802.11 EDCA Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015.

[8] Shestakevych, T., Pasichnyk, V., Kunanets, N., Medykovskyy, M., Antonyuk, N. The content web-accessibility of information and technology support in a complex system of educational and social inclusion. IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 - Proceedings, 2018, pp.102-108.

[9] https://www.maplesoft.com

[10] Oleksiv, I., Izonin, I., Kharchuk, V., Tkachenko, R., Doroshenko, A.: Identification of IT Sector Stakeholder's Requirements to Masters Program in Information System in Lviv Region. In: Ermolayev, V., Suárez-Figueroa, M. C., Ławrynowicz, A., Palma, R., Yakovyna, V., Mayr, H. C., Nikitchenko, M., and Spivakovsky, A. (Eds.): ICT in Education, Research and Industrial Applications. Proc. 14-th Int. Conf. ICTERI 2018. Volume I: Main Conference. Kyiv, Ukraine, May 14-17, pp.112-120 (2018)

[11] Shakhovska, N., Vovk, O., Hasko, R., & Kryvenchuk, Y. (2017, September). The method of big data processing for distance educational system. In Conference on Computer Science and Information Technologies (pp. 461-473). Springer, Cham.