Communication in distributed simulation environment

EGILS GINTERS, ARTIS SILINS
Sociotechnical Systems Engineering Institute
Vidzeme University College
4 Cesu Street, Valmiera LV-4200
LATVIA
http://www.va.lv

JANIS ANDRUSAITIS
Ligatne Natural Trail
Baznicas Street 3, Sigulda LV-2150
LATVIA
http://www.gnp.gov.lv

Abstract: - Growing complexity of the systems determines the important rule of the simulation, because specification possibilities of classical analytical methods are quite limited. The analysis and operation of the real sociotechnical systems ask for cooperation among experts of technical and social (environmental) systems. Serious amount of simulation tools and simulators are created for modelling technical as well as social systems, nevertheless designing of heterogeneous and distributed simulation models is still difficult. One of the reasons is lack of suitable techniques for communications and synchronization of the various models inside the joint systems model. The authors offer the CORBA solution for communication inside distributed simulation model of Ligatne Natural Trails - an integrated tourism object for recreation and ecosystems research.

Key-Words: - Agent-based simulation, Discrete event systems, Distributed simulation, Communication environment, Sociotechnical systems, EXTEND, NetLogo, CORBA, HLA

1 Introduction

Growing variety and complexity of systems determines the important rule of the simulation, because specification possibilities of classical analytical methods are quite limited. Moreover, solutions of designed models in real time are close to problematic.

Changes in thinking and understanding supported by the growing computational performance allow approximating the actors of engineering and social knowledge step by step. Although with displeasure, they must confess that analysis and operation of the real sociotechnical systems, especially stochastic, cannot be substantial without equal notice of both technical factors and social (environmental, biologic) phenomena.

The technical systems are mostly closed, but self-organization and changes in the structure of the model during the session are not typical for them. Unlike the social systems, whose are cognitive and the result of the activity can depend on the new knowledge obtained during the previous iteration and huge amount of important perturbations and inputs. Serious amount of simulation tools and simulators are created for modelling technical as well as social systems allowing to design continuous, discrete, time-driven, event-driven and other models, nevertheless designing of heterogeneous simulation models is still difficult. One of the reasons is lack of suitable techniques for communications and synchronization of the various models inside the joint systems model.

2 Simulation of the Ligatne Natural Trails

The Gauja National Park (Latvia) was founded in 1973 [1]. It focuses mainly on nature protection and educational and leisure tourism. The Ligatne Natural Trails was founded in 1975 as an integrated tourism object for recreation and ecosystems research giving possibility for introduction with flora and fauna, which is typical for Latvian regions. The Ligatne Natural Trails is a part of the Gauja National Park. There exist different routes ensuring movement the visitors either by foot or by car. The total amount of visitors and cars are permanently growing. The problem mentioned above asks for expanding of parking and traffic dispatching. Permanently growing amount of visitors makes pressure on functioning of ecosystem and regeneration possibilities of lawns in sightseeing places. Therefore, potential workload forecasting of the tourism object, which depending of tourists flows intensity, becomes important. In 2006 two separate simulation models were elaborated [2] (see Fig.1).

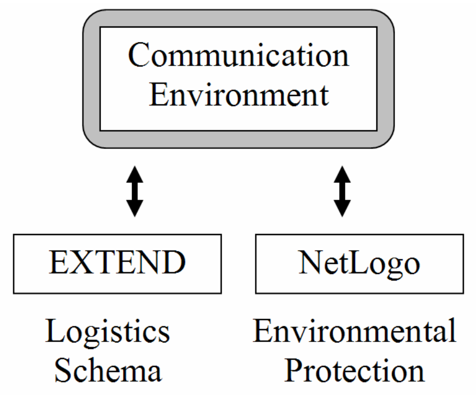

Fig. 1. Structural model of simulation system of the Ligatne Natural Trails

First - for throughput analysis of the trails and scenic routes, and planning the parking places. Functional model in this case was described as discrete event system and was simulated in EXTEND [3]. The second model was provided for estimation of other critical resource as capacity and quality of the sight place. Under the quality we understood the capabilities of regenerating of the nature resources (lawn, for instance) in given area influenced by the flow of the tourists with predefined intensity. In this case agent based approach was used and NetLogo [4] environment was selected. Only one problem still remained related with communication and synchronizing the separate models located on different places and operated in the real time.

EXTEND [3] is modelling environment supporting modelling, analysis and optimisation of the discrete-event and continuing processes. The environment is relatively open and well-linked with MS Office. It is possible to create user's component's using embedded ModL language, which is similar to C language. EXTEND allows the communications with ODBC, ActiveX and C classes.

NetLogo [4] is multi-platform multi-agent programming and modelling environment designed mostly for analysis of environmental systems and social phenomena. NetLogo is created in Java ensuring the platform independent environment, possibilities of object-oriented structuring and good programming options. NetLogo allows designing different extensions for communication with outer environment.

To ensure interoperability and information exchange among the models the convenient communication environment is necessary.

3 Communication environments

Some communication environments and algorithms of collaboration of the distributed systems and agents were analysed to find more convenient of them by the following criterions: performance of the operation, less-consuming contribution for connecting the separate models to the communication environment and maintenance possibilities, and support of the solution. The following environments have been selected:

Aggregate Level Simulation Protocol (ALSP) [5] – both software and a protocol, is used by the United States military to link analytic and training simulations. The main components of ALSP are Infrastructure Software for distributed runtime simulation support and management and Interface – the set of data exchange message protocols for interaction among objects represented in different simulations. Really ALSP is predecessor of the HLA (High Level Architecture) [6];

Common Object Request Broker Architecture (CORBA) [7] – the standard elaborated for joint communication the programmes written in different languages. To achieve this aim the interfaces of the modules must be specified in conformity with the requirements of the Interface Definition Language (IDL). IDL specifications are accessible for C, C++, Java, COBOL, Smalltalk, Ada, Lisp and Python, which made this mechanism well suited enough. Nevertheless, the CORBA environment asks for centralised synchronisation that made distributed simulation systems vulnerable, but it is important only in some specific cases;

The Foundation for Physical Intelligent Agents (FIPA) specifications [8] are a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. Many of the ideas originated and developed in FIPA are now coming into sharp focus in new generations of Web/Internet technology and related specifications. FIPA standards for agents and multi-agent systems were officially accepted by the IEEE in 2005. It was decided to look on those specifications into the wider context of software development and integrate with non-agent technologies. In close future this approach would became one of more exciting ways for communication of the different systems.

High Level Architecture (HLA) [6] is concept of the architecture for distributed simulation systems. HLA ensure interoperability and reuse among simulations. It consists of Rules that simulations (federates) must follow to achieve proper interaction during a federation execution; Object Model Template (OMT) that defines the format for specifying the set of common objects used by a federation (federation object model), their attributes, and relationships among them; Interface Specification (IFSpec), which provides interface to the Run-Time Infrastructure (RTI), which can be distributed and ties together federates during model execution. The distributed time management can be done, because all federates' nodes directly undertake synchronization roles. Therefore, the total simulation is more quickly and the system is safer, unfortunately, implementation is more complex and laborious.

In addition, each of the environments and the protocols have their own lacks, nevertheless, respecting the criterions of the selection declared above the most convenient environment would be CORBA.

4 CORBA use for simulators collaboration

The CORBA architecture is shown in Fig. 2, where the following items are mentioned [9]:

Object – a programming entity that consists of an identity, an interface, and an implementation, which is known as a Servant;

Servant - an implementation entity that defines the operations supported by CORBA IDL interface. Servants can be written in a variety of languages, including C, C++, Java, Smalltalk, and Ada;

Client - the program that invokes an operation on an object implementation;

Object Request Broker (ORB) provides a mechanism for transparently communicating client requests to target object implementations. The ORB simplifies distributed programming by decoupling the client from the details of the method invocations. When a client invokes an operation, the ORB is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object, and returning any response to the caller;

ORB Interface - a logical entity that may be implemented as one or more processes or a set of libraries. To decouple applications from implementation details, the CORBA specification defines an abstract interface for an ORB;

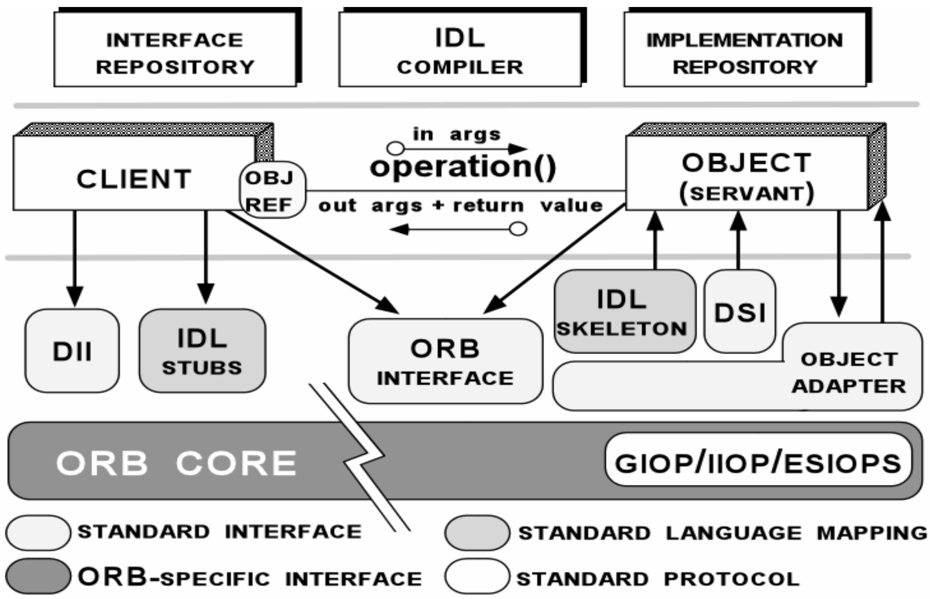

Fig. 2. CORBA Object Request Broker architecture [9]

CORBA IDL stubs and skeletons - the „glue" between the client and server applications, respectively, and the ORB. The transformation between CORBA IDL definitions and the target programming language is automated by a CORBA IDL compiler;

Dynamic Invocation Interface (DII) allows a client to directly access the underlying request mechanisms provided by an ORB. Applications use the DII to dynamically issue requests to objects without requiring IDL interface-specific stubs to be linked in;

Dynamic Skeleton Interface (DSI) allows an ORB to deliver requests to an object implementation that does not have compile-time knowledge of the type of the object it is implementing. DSI is the server side's analogue to the client side's DII;

Object Adapter - assists the ORB with delivering requests to the object and with activating the object. An object adapter associates object implementations with the ORB;

Internet Inter-ORB Protocol (IIOP) is protocol provided for interoperability between CORBA-compliant ORBs. IIOP is the TCP/IP transport mapping of a *General Inter-ORB Protocol* (GIOP). IIOP enables requests to be sent to networked objects. The CORBA interoperability architecture also accommodates communication using optional *Environment-Specific IOPs (ESIOPS).*

Designing of distributed simulation model of the Ligatne Natural Trails therefore involved three steps (see Fig. 3):

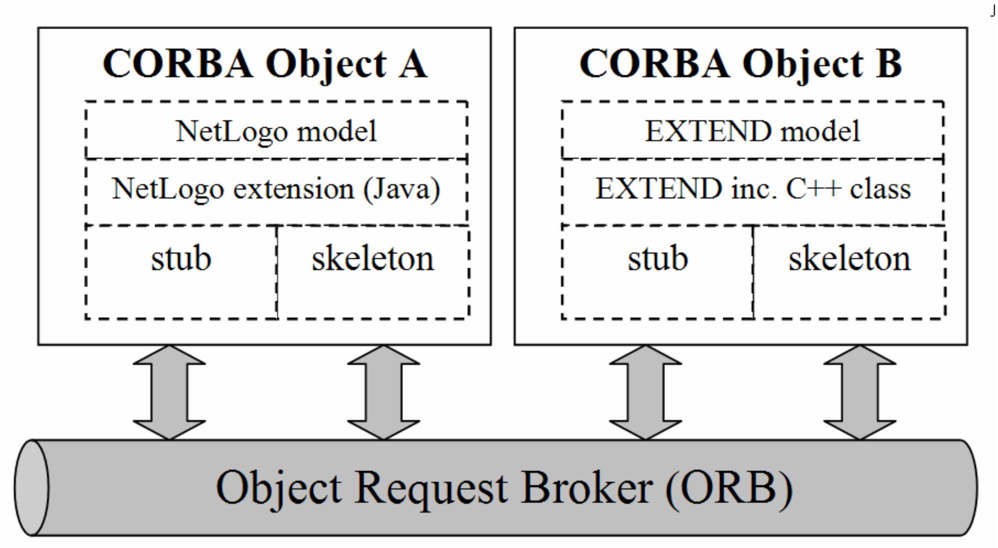

Fig. 3. NetLogo and EXTEND integration via CORBA

- The specific extension for NetLogo environment for communication between models and outer space was designed. It was done in JAVA language in conformity with NetLogo requirements;
- The similar extension was designed for discrete-event simulation environment Extend. It was done in C++, because EXTEND supports the implementation of C++ classes;
- Integration of the constructions designed above in CORBA architecture.

In Fig. 3 two CORBA objects are shown marked by A and B and involving NetLogo and EXTEND models discussed previously [2].

Each object has "stub" and "skeleton", where "stub" is used for client functions support giving possibilities for connecting to the server.

Otherwise, "skeleton" allows to the object the server's functions implementation granting answering to the requests. NetLogo extension in Java and EXTEND extension in C++ ensure collaboration of the simulators with CORBA ORB environment.

5 Conclusion

Sociotechnical systems are manageable systems respecting human and/or biologic and/or environmental factor role, mutual assistance, and influence on joint functioning of the both systems – technical and social [10].

Serious amount of simulation tools and simulators are created for modelling technical (EXTEND, ARENA, Witness, AutoMod, Simul8, STELLA etc.) [11] as well as social systems (NetLogo, AgentSheets, SWARM, RePast, MIMOSA etc.) [12], nevertheless designing of heterogeneous simulation models is still difficult. One of the reasons is lack of suitable techniques for communications and synchronization of the various models inside the joint systems model.

Some different protocols and methods (CORBA, HLA, FIPA, ALSP and other) [13, 14] exist for elaboration the communication environment, unfortunately any of it has their own disadvantages and right selection is still problematic.

Because no solutions are useable without programming knowledge then the authors offer the CORBA (Common Object Request Broker Architecture) [15] for integration of the separate models operating inside the goal model of the heterogeneous system. The authors consider that given approach, which is practically checked in the Ligatne Natural Trails simulation system, is more convenient by criterions of the performance of operating, possibilities for the maintenance and support, and labour-consuming as well.

6 Acknowledgement

The presented activity is supported by the agreement No. 2007/1-17/26 of the Latvian Ministry of Education and Science "COMMUNICATION ENVIRONMENT OF HYBRID SIMULATION SYSTEMS".

References:

[1] Gauja National Park, http://www.gnp.lv/eng/index.htm, 14.09.2006.

[2] E. Ginters and A. Silins, Multi-level approach for environmental systems modelling in the Ligatne Natural Trails, in *WSEAS Transactions on Systems*. ISSN 1109-2777. Issue 4, Volume 6, April 2007, 795-801.

[3] D. Krahl, The EXTEND Simulation Environment, in *Proceedings of the 2002 Winter Simulation Conference*/ E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., 2002.

[4] U. Wilensky, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999 and http://ccl.northwestern.edu/netlogo/, 11.09.2007.

[5] What is ALSP? http://ms.ie.org/alsp/questions/What%20is/index.html, 11.09.2007.

[6] K. Perumalla, Tutorial. Handling Time Management Under the High Level Architecture, 2006, http://www.ornl.gov/~2ip/doc/perumalla-tutorialslides-iitsec06.pdf, 12.09.2007.

[7] Welcome to the CORBA Directory. http://corba-directory.omg.org/, 11.09.2007.

[8] The Foundation for Physical Intelligent Agents. http://www.fipa.org, 12.09.2007.

[9] Overview of CORBA, http://www.cs.wustl.edu/~schmidt/corba-overview.html, 12.09.2007.

[10] E. Ginters and A. Silins, Modelling tools for environmental systems analysis in the Gauja National Park, in *Proceedings of 6th WSEAS International Conference on Applied Computer Science (ACS'06)*, ISSN 1790-5117, ISBN 960-8457-57-2, Tenerife, Puerto de la Cruz, Spain, December 16-18, 2006, 194-198.

[11] Verbraeck, Chapter 8. Simulation, in E. Ginters (ed). *Logistic Information Systems*, Part 2. Jumi, Riga, Latvia, ISBN 9948-30-021-8. 2002, p. 100-133.

[12] N. Gilbert and K. G. Troitzsch, *Simulation for the Social Scientist*, Open University Press, 2 edition, ISBN 978-0335216017, 2006, P.312.

[13] S. Strassburger, The road to COTS-interoperability: from generic HLA-interfaces towards plug-and-play capabilities, in *Proceedings of the 37th conference on Winter simulation*, December 03-06, 2006, Monterey, California.

[14] A. Verbraeck, Component-based Distributed Simulations. The Way Forward?, in *Proceedings 18th Workshop on Parallel and Distributed Computer Simulation*. (Kufstein, Austria, 16-19 May 2004), IEEE Computer Society Press, Los Alamitos, CA, 2004. pp. 141-148.

[15] F. Bolton, *Pure CORBA*, Sams, ISBN 978-0672318122, 2001, P. 944.