SpaceWire-RT Standard Simulation Models

Ilya Korobkov, Irina Lavrovskaya, Valentin Olenev
St-Petersburg State University of Aerospace Instrumentation
St-Petersburg, Russia
{Ilya.Korobkov, Irina.Lavrovskaya, Valentin.Olenev}@guap.ru

Abstract

Modern and perspective highly capable on-board networks for scientific space missions require responsiveness, determinism, robustness and durability. The SpaceWire-RT research programme aims to create communications network technology, suitable for these fundamental requirements of demanding applications. The paper gives an overview of activities in the FP7 SpaceWire-RT project consisting in the development of the new onboard communication standard – SpaceWire-RT that satisfies all requirements of space applications. The main focus of the paper is the investigation of SpaceWire-RT standard by means of the SpaceWire-RT protocol simulation. The implemented SDL and SystemC models gave an ability to validate the current version of SpaceWire-RT standard and consequently produce the new version of the specification.

Index Terms: Simulation, SpaceWire-RT, Communication protocols, SDL, SystemC.

I. Introduction

Simulation plays one of the most important roles in a process of communication protocols design. It is used to validate the specification and to find ambiguities and inconsistencies [1]. So this was done for the newest onboard communication standard – SpaceWire-RT [2].

SpaceWire-RT is a critical component technology for future spacecraft avionics and payload data-handling. The creation of this technology will substantially strengthen collaborative bonds between the Russian and European organisations involved in the research, and lead to technology of vital importance for future space missions.

The SpaceWire-RT project aims to provide a flexible, robust, responsive, deterministic and durable standard network technology for spacecraft avionics, i.e. spacecraft onboard data-handling and control electronics.

The paper presents the SpaceWire-RT standard and gives an overview of simulation work that has been done in the scope of the FP7 SpaceWire-RT project. The description of SDL point-to-point model and SystemC network model that were implemented for the SpaceWire-RT simulation and validation is followed by results of the simulation and research.

II. Overview of SpaceWire-RT Communication Standard

Aerospace industry is one of the most rapidly growing areas in terms of communication protocols development. Avionics and robotics impose requirements on network responsiveness and determinism. The increasing international collaboration on scientific and Earth observation spacecraft requires standard network technology where a component developed by one nation will interoperate effectively with equipment developed by another nation. SpaceWire-RT is the newest onboard communication protocol standard, which aims to fulfill these demanding requirements with a flexible, robust, responsive, deterministic and durable standard network technology that is able to support both avionics and payload data-handling applications [2].

SpaceWire-RT aims to cover many on board communications applications from low to very high data-rate networks. This is a critical component technology for future spacecraft avionics and payloads.

SpaceWire-RT will:

- use virtual channel concepts to provide a variety of QoS;
- provide broadcast and multicast capability;
- increase performance;
- provide low latency message delivery;
- include extremely low latency time and out-of band signalling mechanisms;
- incorporate novel fault detection, isolation and recovery methods;
- make the network fully responsible for information transfer;
- decouple application and data transfer;
- implement appropriate communication mechanisms in relatively simple hardware.

SpaceWire-RT standard is based on the SpaceFibre technology [2].

SpaceFibre is a very high-speed serial data-link, which is intended for use in data-handling networks for high data-rate payloads. SpaceFibre is able to operate over fibre optic and copper cable and support data rates of 2 Gbit/s in the near future and up to 20 Gbit/s long-term. SpaceFibre will provide a coherent quality of service mechanism able to support best effort, bandwidth reserved, scheduled and priority based qualities of service. It will substantially improve the fault detection, isolation and recovery (FDIR) capability of SpaceWire [2], [3], [4].

SpaceFibre provides robust, long distance communications for launcher applications and will support avionics applications with deterministic delivery constraints through the use of virtual channels. SpaceFibre enables a common onboard infrastructure to be used across many different mission applications resulting in cost reduction and design reusability. SpaceFibre can run over fibre optic or copper cables [2].

SpaceFibre has many advantages in comparison with SpaceWire [2], [3], [4]:

- It uses fewer wires reducing cable mass.
- It operates at data rates of 2 Gbits/s and potentially higher.
- It uses matched impedance connectors.
- The size of all the characters are the same (32-bits).
- Parity coverage is per character.
- It uses a DC balanced encoding scheme.
- It provides simple capacitive, magnetic, or optical galvanic isolation.
- The initialization protocol is base on a double handshake.

An overview of the SpaceWire-RT architecture is provided in Fig. 1.

There are twelve conceptual layers to the SpaceWire-RT:

- ***Network:*** responsible for routing SpaceWire-RT packets over a SpaceWire-RT network, comprising SpaceWire-RT routing switches, SpaceWire-RT links, and SpaceWire-RT nodes. Also responsible for validating and broadcasting SpaceWire-RT broadcast messages over a SpaceWire-RT network.

- ***Packet:*** responsible for forming data to be sent over a SpaceWire-RT link into packets, which have the same format as SpaceWire packets.
- ***Virtual Channel:*** responsible for quality of service and flow control over the SpaceWire-RT link.
- ***Broadcast:*** responsible for sending short broadcast messages across a SpaceWire-RT link and for receiving those messages.
- ***Framing:*** responsible for framing SpaceWire-RT packet data, broadcast messages and FCTs to be sent over the SpaceWire-RT link. It is also responsible for scrambling SpaceWire packet data for EMC mitigation purposes.
- ***Retry:*** responsible for recovering from transient errors on the SpaceWire-RT link, and for reporting errors and link failure. Detects missing and out of sequence frames.

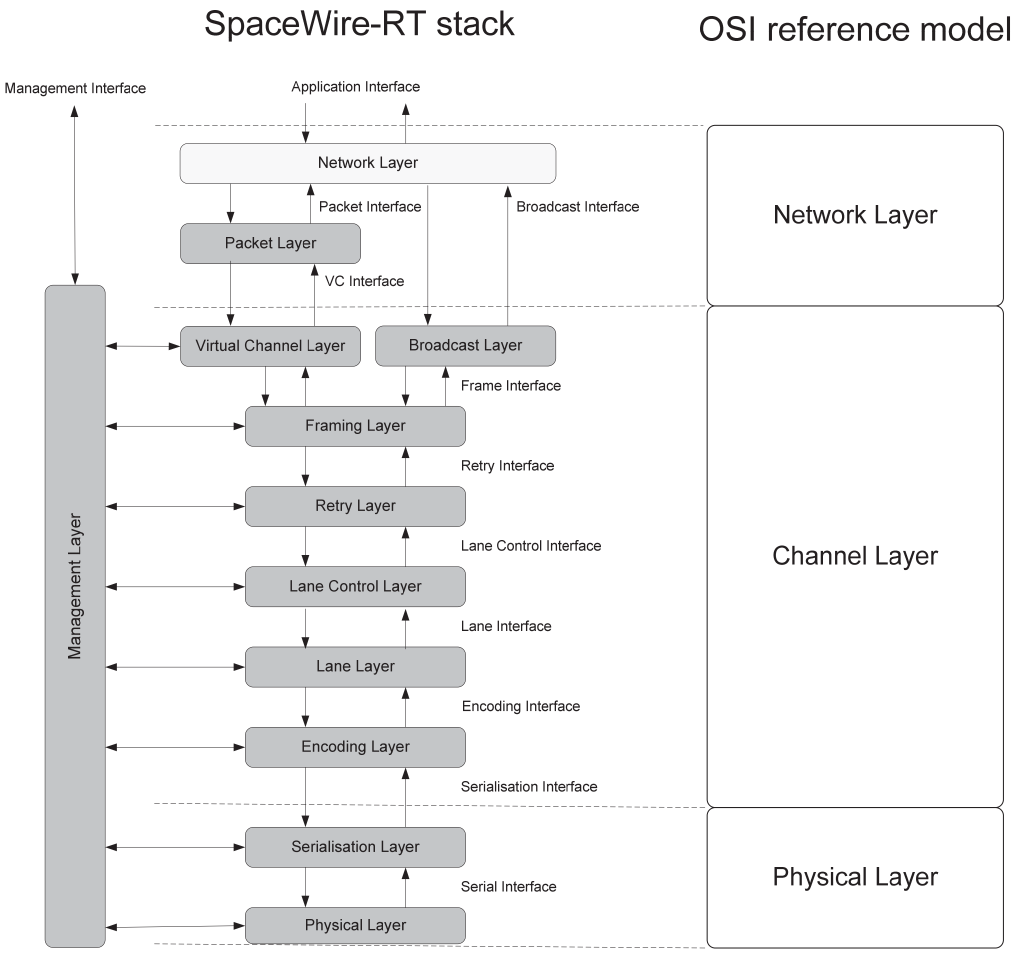

Fig. 1. The SpaceWire-RT layered architecture

- ***Lane Control:*** responsible for operating several SpaceWire-RT lanes in parallel to provide a higher data throughput and to provide redundancy with graceful degradation.

- ***Lane:*** responsible for initialising the lane, detecting lane errors and re-initialising the lane after an error has been detected.
- ***Encoding:*** responsible for encoding the data and control words into a suitable form for sending over the SpaceWire-RT link and decoding received data and control words. Uses 8B/10B encoding.
- ***Serialisation:*** responsible for serialising and de-serialising encoded data and control words for sending and receiving over the serial interfaces.
- ***Physical:*** responsible for sending the SpaceWire-RT information over the physical media used in SpaceWire-RT: fibre optic, Current Mode Logic (CML) and Low Voltage Differential Signalling (LVDS). Fibre Channel copper medium may also be added in future.
- ***Management:*** responsible for configuring, controlling and monitoring the status of the various layers of the SpaceWire-RT protocol stack. This can be done by a local or remote network management application [2].

SUAI modeling team developed SpaceWire-RT simulation models in SDL [5] and SystemC [6] modeling languages. An SDL model is used for validation of SpaceWire-RT mechanisms. A SystemC model is designed for network communication mechanisms testing.

III. SPACEWIRE-RT P2P MODEL IN SDL

SDL (Specification and Description Language) is a language for unambiguous specification and description of the telecommunication systems behavior. The SDL model covers the following five main aspects: structure, communication, behavior, data and inheritance. SDL language is intended for description of structure and operation of the distributed real-time systems. Writing an SDL model on the basis of the specification is itself a test of the specification for completeness and unambiguousness. As a result the consistent readable textual description and formalised specification in SDL are produced [7].

SDL language was used for SpaceWire-RT specification and simulation on a per layer basis as the most reasonable solution. The SDL model formally describes all mechanisms, interactions and functionality which are stated in the SpaceWire-RT specification [7].

The SDL model implements all layers of the SpaceWire-RT protocol stack (excepting Serialisation Layer). Fig. 2 shows the general structure of the SpaceWire-RT node in SDL.

According to the figure above the SDL model describes the internal mechanisms and functionality of the layers starting from the Encoding Layer and up to the Virtual Channel Layer. Each pair of adjacent layers communicates via a special interface between them, which is called a Service Access Point (SAP). All SAPs are defined as sets of service primitives which are specific for each layer.

Simulation and investigation was done in two steps. First of all we performed verification of the SpaceWire-RT protocol stack by simulation in IBM Rational SDL Suite. The test system was represented by the two SpaceWire-RT nodes communicating through the Serialisation Layer channel. Configuration and generation of test sequences was performed by a special Test Engine. This simulation gave an ability to check all internal mechanisms of investigated layers and verify them.

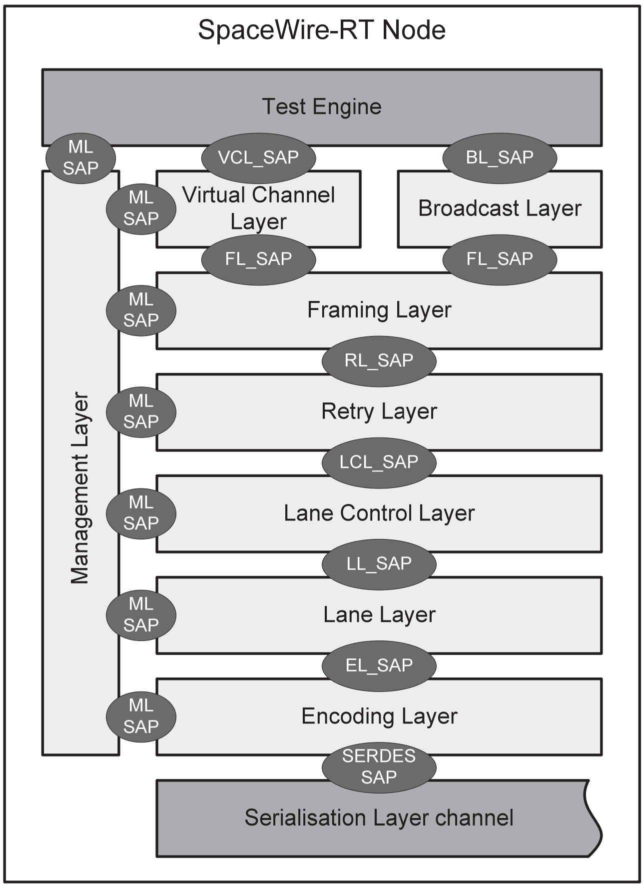

Fig. 2. General structure of the SpaceWire-RT node

The second step was validation of the SpaceWire-RT protocol stack by load testing by means of simulation within an SDL/SystemC tester. The SDL/SystemC tester provides a possibility for simulation of a point-to-point interconnection between two nodes, implemented in SDL and communicating via a channel. The tester is a flexible tool for setting different configurations, generating various test sequences and gathering statistics.

The general structure of the SpaceWire-RT SDL/SystemC tester is given in Fig. 3.

The SystemC Test Engine provides facilities for creation of different complicated test sequences with different configurations and efficient logging of the events in the model.

In order to implement interconnection between the SpaceWire-RT SDL model and the test environment an SDL/SystemC co-modeling approach was used [8]. This approach assumes that special wrappers (SDL/SystemC up wrapper and SDL/SystemC low wrapper) should be implemented for conversion of data from the SDL representation to the SystemC representation and vice-versa. The wrapper receives SystemC data, converts it into the SDL signals and sends to the SDL model via the correspondent SAP. Thus, the up wrapper is responsible for communication of test engine and SDL model and the low wrapper is responsible for communication of the SDL model and the channel.

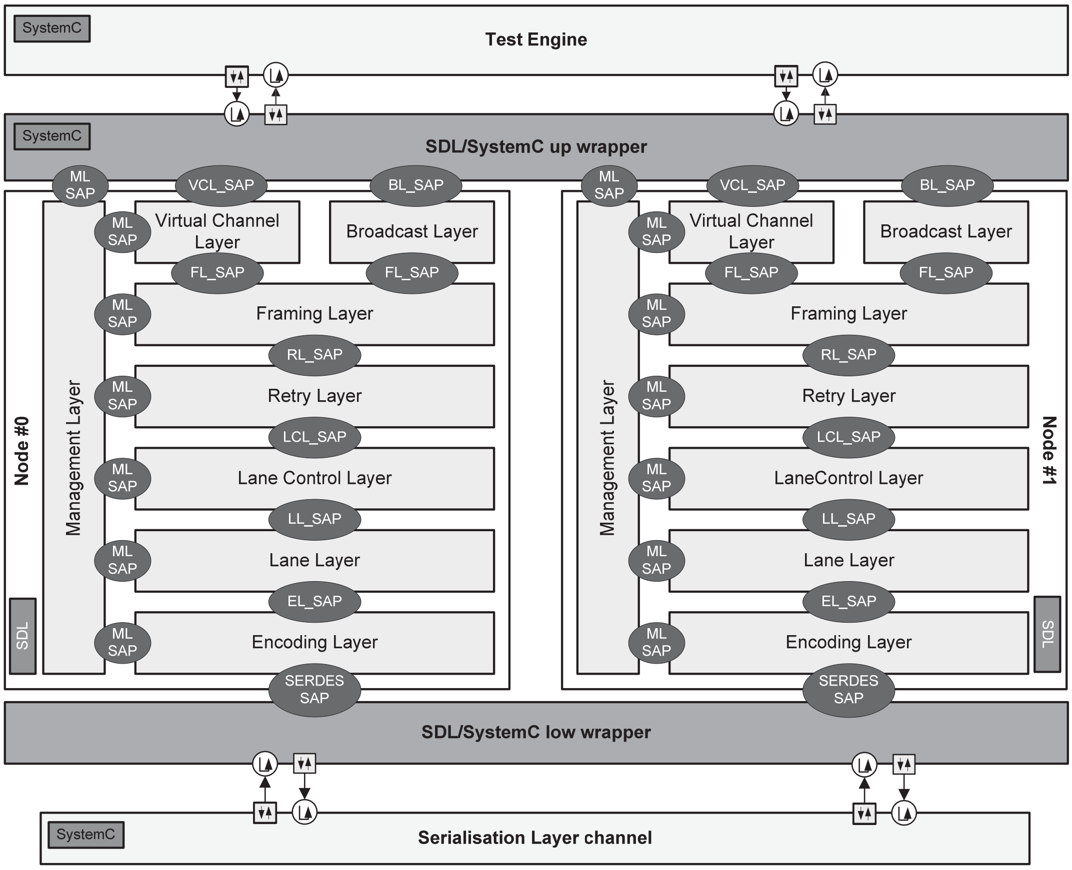

Fig. 3. General structure of the SpaceWire-RT SDL/SystemC tester

The target SDL model of the whole SpaceWire-RT protocol stack can be used for checking, how all mechanisms operate in common in one node by means of simulation in the IBM Rational Tool and by means of simulation within the SDL/SystemC tester. This way, the SDL layered SpaceWire-RT model can be used for validation of consistency of the specification and checking of functional requirements, defined for the standard. The main advantage of this model is that it is implemented in a formal high-level language. This model can be used for further investigation of SpaceWire-RT technology because any changes in new versions of the standard could be applied to the SDL specification without any changes in the test environment.

IV. SPACEWIRE-RT NETWORK MODEL IN SYSTEMC

The SystemC modeling is one of the most efficient and widely used methods for studying, analysis and constructing multi-component systems, such as stacks of protocols, embedded networks of a large number of nodes, systems-on-chip, networks-on-chip, etc [9], [10].

SystemC is a set of C++ classes and macros that provide an event-driven simulation engine. It is specifically designed for modeling parallel systems. This library allows describing multi-component systems and program components, and modeling their operation. By using the internal mechanism of events it allows to model operations distributed in time of the modeled system [6], [9].

The aim of the network SpaceWire-RT model development is to simulate communication of devices (switches and nodes) via the SpaceWire-RT links. In the SystemC network model some interactions of components and processes inside the device (e.g. between levels of a stack) could be not considered, because the model is primarily focused on the mechanism of devices' communications, such as transfer of packets, routing and performance characteristics of the network [7].

The SpaceWire-RT network model consists of the following SystemC modules:

- SpaceWire-RT stack model, which provides main functions of SpaceWire-RT.
- SpaceWire-RT node model.
- SpaceWire-RT switch model.

The SpaceWire-RT stack model is a part of the node and the switch models. For these models it is possible to set different parameters like: a data transmission speed (Gbps), a number of nodes and switches, size and amount of packets, a destination address for a particular packet, a time delay and a routing table for the switch, a number of ports in the switch, etc.

The SpaceWire-RT network model contains a number of nodes and switches. It could contain no switches so the model would be point-to-point. It gave an ability to simulate operation of the various number of devices in a network with different topologies: point-to-point, tree and circular.

Using a point-to-point configuration we had an ability to check correctness and consistency of the SpaceWire-RT stack specification. An example of the point-to-point configuration is shown in Fig. 4.

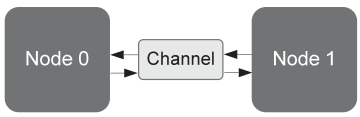

Fig. 4. Point-to-point network configuration

For testing of network mechanisms we used the mixed configuration, which is a combination of tree and circular topologies. Mixed configuration gave an opportunity to check the following network parameters: latency for different packet sizes, reliability of data transfer with specific BER (Bit Error Rate), various QoS (Quality of Service), fault packet detection and identification, failure and fault tolerance of a network (deadlock and babbling idiot), broadcast and multi-cast, path and logical addressing. An example of the used mixed network configuration is given in Fig. 5.

V. SIMULATION RESULTS

Beforementioned SDL and SystemC models were used for the scientific studies and research. SpaceWire-RT standard was checked on conformance to the Russian and European industry requirements and also on existence of inconsistencies and ambiguities. Some of the most important results of our research are given below.

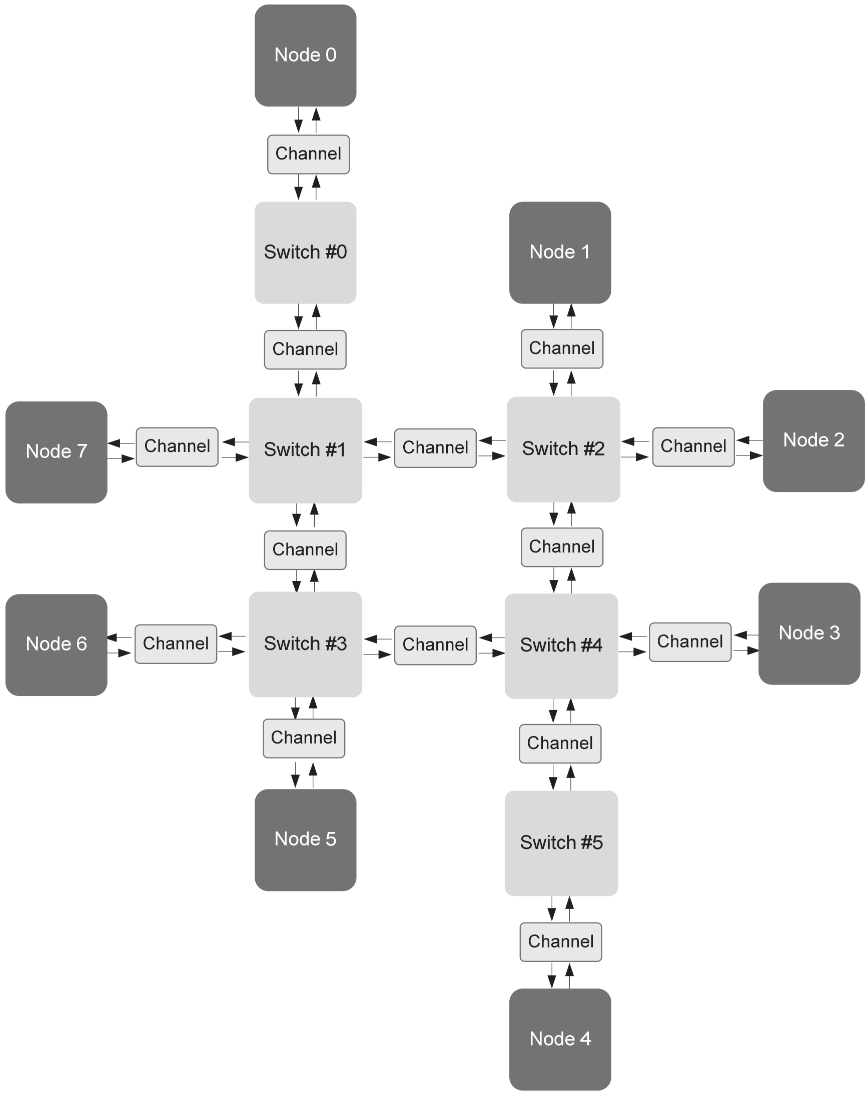

Fig. 5. An example of a mixed network configuration

- SpaceWire-RT packets (length up to 32 MBytes) are successfully transmitted over a single SpaceWire-RT link.
- Broadcast messages transmission is successfully implemented in a standard. However, the latency for the broadcast messages does not fit the requirement of 100 ns even if we have only one switch between the nodes. So the requirement is too strict for the current technology.
- SpaceWire-RT provides a capability for reliable data delivery.
- Scheduling mechanism of SpaceWire-RT successfully provides deterministic data delivery.
- Additional functionality should be added to the current retry mechanism, because automatic acknowledgement mechanism at the Retry Layer is not configurable, but it should be according to the requirements of the industry.

- Automatic fault detection at different layers of the stack is successfully implemented.
- An occurrence of a deadlock for one of the virtual channels does not stop the data transmission for the other virtual channels.
- An occurrence of a babbling idiot for one of the virtual channels does not stop the data transmission for the other virtual channels. The transmission speed is highly dependent on a priority level of the virtual channel used by a babbling idiot.
- Multi-path and multi-cast transmission is supported and checked on a network model.
- SpaceWire-RT does not provide the mechanism of the broadcast messages discard in switches during the repeated transmission over a network with a circular structure.

The detailed information of the testing and simulation results would be available after the finish of the project.

V. Conclusion

The paper describes the new SpaceWire-RT technology and its validation by means of SDL and SystemC models. During the simulation our team found a number of inconsistencies in the specification and proposed some solutions and additional mechanisms to solve them. The new version of the SpaceWire-RT standard is produced and it is based on the simulation impact. The latest news and results of the project are available on our website http://www.spacewire-rt.org.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° 263148.

References

[1] A. Jantsch, “Modeling Embedded Systems and SoCs”, *Morgan Kaufmann Publishers, Stockholm*, 2004.
[2] S. Parkes, “SpaceWire-RT Outline Specification, version 2.1” , *University of Dundee*, 6th September 2012.
[3] ESA (European Space Agency), standard ECSS-E-50-12A, “Space engineering. SpaceWire – Links, nodes, routers and networks. European cooperation for space standardization”, *ESA Publications Division ESTEC, Noordwijk, The Netherlands*, 2003.
[4] Y. Sheynin, T. Solokhina, Y. Petrichkovitch “SpaceWire technology for the parallel systems and onboard distributed systems”, *ELVEES*, 2006, http://multicore.ru/fileadmin/user_upload/mc/publish/SpW-part1.pdf.
[5] International Telecommunication Union, “Recommendation Z100: Specification and Description Language (SDL)”, 2007.
[6] Open SystemC Initiative (OSCI), “IEEE 1666™-2005 Standard for SystemC”, 2005.
[7] Y. Sheynin, E. Suvorova, V. Olenev, I. Lavrovskaya, “D3.1 SpaceWire-RT Simulation and Validation Plan”, *Saint-Petersburg State University of Aerospace Instrumentation*, 3rd October 2012.
[8] S. Balandin, M. Gillet, I. Lavrovskaya, V. Olenev, A. Rabin, A. Stepanov, “Co-Modeling of Embedded Networks Using SystemC and SDL”, *International Journal of Embedded and Real-Time Communication Systems (IJERTCS), IGI Global*, pp. 24-49, 2011.
[9] J. Gipper, “SystemC the SoC system-level modeling language. Embedded computing Design”. 2007.
[10] V. Olenev, “Different approaches for the stacks of protocols SystemC modelling analysis”, *Proceedings of the Saint-Petersburg University of Aerospace Instrumentation scientific conference, Saint-Petersburg University of Aerospace Instrumentation (SUAI), Saint-Petersburg*, pp. 112-113, 2009.