Working with Open International Standards

Institute for System Programming
*of the Russian Academy of Sciences (ISPRAS) **

ISP RAS has gained extensive experience in processing international standards, both open[1] and closed. The primary focus of this paper are standards that specify structured communication between systems such as programming interfaces, message exchanges or processing files with instructions in a programming language. The following list presents standards of this category that ISP RAS involved in its projects. In this list open and semi-open standards are printed in the straight font and closed standards are printed *in italics*:

- API specifications: POSIX [1], Linux Standard Base [2], *ARINC* [3];
- language specifications: C [4], C++ [5], Java [6], C# [7], TTCN-3 [8];
- multi-media specifications: *MPEG-2* [9], *MPEG-4* [10], *MPEG-21* [11];
- protocol stack specification –TCP/IP/IPv6 stack [12-14];
- security system specifications: IPsec [15, 16], *MPEG -2 IPMP* [9].

Conformance Test Suite Development

Historically first ISP RAS projects that involved Open Standards were focused on conformance test suites development. ISP RAS has invented the UniTESK[17] technology to automate conformance test suite development.

According to UniTESK the development process for a conformance test suite is composed of several steps:

1. standard study and functional requirements elicitation;
2. rewriting elicited requirements using formal description language;
3. test scenario development for automated check of conformance between an implementation and the formal specification;
4. test execution and test report generation.

The distinguishing feature of the UniTESK technology is using formal models of standards that provide the basis for automatic conformance check. The most of modern standards are written in English and do not use formal notations to settle operational semantics. English descriptions are too ambiguous to serve as basis for automated test procedures. This leads to necessity to transform requirements stated in standards into a formal model of the corresponding

[1] Open standards attracts experts communities who work with the standard in an open and collaborative fashion. Community openness is the key feature of such standards that distinguishes them from the closed ones: the latter are being developed by groups of experts as well, but those groups are relatively closed and do not provide possibility to join them in a free manner. Major differences between open and closed standards are:

1. The process of open standard development is open – there are web conferences or mail lists where everybody is invited to publish their proposals or comments about the standard.
2. There is free access to texts of open standards and supporting materials. There are no limitations (especially financial) to gain such access.

In some cases standards have mixed models of open and closed processes.

standard. The formal model is a rigor, unambiguous and machine-readable interpretation of the standard.

Since 2000 ISP RAS carries projects on conformance test suite development for TCP/IP/IPv6 stack of protocols for Internet: IP, IPv6, Mobile IPv6, IP security. The resulting test suites were used to test both commercial and open source IPv6 implementations.

In 2005 ISP RAS initiated a project on development a conformance test suite for POSIX standard. POSIX started as a closed standard in early 1990-s but 10 year ago it became an open standard under control of Austin Group community. While working on the test suite, ISP RAS team contacted the Austin Group to clarify some statements of the standard and to report identified defects and proposed changes. The test suite was used to test a real-time operating system and forms the core of the test suite for the Linux Standard Base.

Currently ISP RAS is involved in conformance and certification test suite development for Linux Standard Base [18]. This project is carried out in close cooperation with the Linux Foundation – the official consortium leading LSB development and promotion.

Since 2006 ISP RAS is working on conformance test suite for ARINC-653. The test suite extends conformance statements from the Part 3 of the standard and covers most of the functional requirements of ARINC-653. The test suite was used to test a real-time avionics operating system.

Standard Improvement and Refinement

Building a formal model of a standard is inseparable from thorough study and analysis of the standard. Such analysis typically reveals ambiguity, contradictions, incompleteness and even errors.

During conformance test suite development for POSIX ISP RAS team performed through standard analysis. It resulted in 25 defect reports to Austin Group: error in function descriptions (such as using undefined constants), contradictions and typos. The study of LSB resulted in 44 defect reported of the similar kind.

For MPEG-2 Intellectual Property Management and Protection subsystem ISP RAS performed a specialized text study focused on the interoperability problems revealing. The project identified about 40 defects in the draft standard including severe problems that might result in breaking interoperability between conforming implementation.

Infrastructure development for standardization process

In 2007 the Linux Foundation ordered ISP RAS to develop a software toolkit to support LSB development process. The toolkit includes LSB Navigator and web-interfaced control system for automated self-certification for LSB conformance.

The Navigator provides tree groups of services:

1. Navigation through LSB elements: functions, components, header files, libraries, etc.
2. Navigation through Linux distributions and applications. The Navigator provides information about libraries set and headers included in each distribution or required by applications.
3. LSB development process support: access to LSB text, tests, coverage and statistics.

The Navigator is being intensively used in the day-to-day work of the Linux Foundation.

Automated certification system includes registration on the LSB site, certification test suite files and automated test results audit system. By the middle of 2008 there are over 40 Linux distributions that passed certification procedures using this automated system.

Implementation and Prototype Development

ISP RAS has developed implementations and prototypes for a number of standards. First a all it refers to programming languages: ISP RAS has developed complete or prototype implementations of C/C++, Java, C#, TTCN-3. ISP RAS has implemented some of IPv6 transitioning mechanisms and provided prototype implementations of MPEG-2 subset.

Cooperation with Standardization Bodies

ISP RAS contacted many standardization bodies in standard-related projects, both open communities and closed workgroups that require performing certain actions to join. Some of these bodies follow a semi-open process – they are open for submitting commenting and proposal and closed for joining the committee.

The following list is some of standardization bodies that ISP RAS worked with:

1. Internet Engineering Task Force (IETF): Internet standards, an open community;
2. OpenGroup: POSIX standards, an open community;
3. The Linux Foundation: Linux Standard Base, an open community;
4. European Telecommunication Standards Institute (ETSI): методология тестирования и язык TTCN-3, a partially open community.
5. The government of China: MPEG-2 and MPEG-2 IPMP standardization, a closed community.

Table 1. Summary table on standard-related activities of ISP RAS.

	API specification	Programming language specification	Stream-processing system specification	Protocol specification	Security system specification
	POSIX, Linux, *ARINC*, JDK	C, C++, Java, C#, TTCN-3	*MPEG-2, MPEG-4, MPEG-21*	TCP/IPv4 /IPv6	IPsec, IPsec v2, *IPMP*
Standard analysis	✓	✓	✓	✓	✓
Standard improvement and refinement	POSIX LSB				IPMP
Conformance test suite development	✓	C, Java	MPEG-2 IPMP	✓	✓
Infrastructure development.	LSB				

	API specification	Programming language specification	Stream-processing system specification	Protocol specification	Security system specification
Prototype or implementation development		C, C#, Java		IPv6 Transitioning	IPMP

References

1. ISO/IEC 9945. Information technology -- Portable Operating System Interface (POSIX). Geneva, Switzerland.
2. The Linux Foundation. Linux Standard Base. [HTML] (http://www.linuxbase.org/)
3. ARINC-653. Avionics Application Software Standard Interface.
4. ISO/IEC 9899. Programming Languages — C. Geneva, Switzerland. 2003.
5. ISO/IEC 14882. Programming Languages — C++. Geneva, Switzerland. 1999.
6. Java Language Specification. Sun Microsystems. 2000. [HTML] (http://java.sun.com/docs/books/jls/second_edition/html/intro.doc.html)
7. ISO/IEC 23270. Information technology — Programming languages — C#.
8. ETSI ES 201 873. Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3.
9. ISO/IEC 13818. Information technology – Generic coding of moving pictures and associated audio information. Geneva, Switzerland.
10. ISO/IEC 14496. Information technology – Coding of audio-visual objects. Geneva, Switzerland.
11. ISO/IEC Technical Report 21000. Information technology – Multimedia framework (MPEG-21). Geneva, Switzerland. 2004.
12. IETF RFC 791. J. Postel. Internet Protocol. IETF, 1981. [TXT] (http://www.ietf.org/rfc/rfc0791.txt).
13. IETF RFC 2460. S. Deering, R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. December 1998. 39 c. [TXT] (http://www.ietf.org/rfc/rfc2460.txt).
14. IETF RFC 793. J. Postel. Transmission Control Protocol. IETF, 1981. 85 c. [TXT] (http://www.ietf.org/rfc/rfc0793.txt)
15. IETF RFC 2401. S. Kent, R. Atkinson. Security Architecture for the Internet Protocol. November 1998. 66 c. [TXT] (http://www.ietf.org/rfc/rfc2401.txt)
16. IETF RFC 4301. S. Kent, K. Seo. Security Architecture for the Internet Protocol. December 2005. 101 c. [TXT] (http://www.ietf.org/rfc/rfc2401.txt)

17. Igor B. Bourdonov, Alexander S. Kossatchev, Victor V. Kuliamin, and Alexander K. Petrenko. UniTesK Test Suite Architecture. In proceedings of FME 2002. [PDF] http://www.unitesk.com/download/papers/unitesk/FME4.pdf

18. Linux Verification Center. [HTML] (http://linuxtesting.org/)