CATALOGUE OF TECHNOLOGIES

2018

CONTENTS

CATALOGUE OF TECHNOLOGIES

ARUTYUN AVETISYAN DOCTOR OF SCIENCE, CORRESPONDING MEMBER OF THE RUSSIAN ACADEMY OF SCIENCES, DIRECTOR OF ISP RAS.

Ivannikov Institute for System Programming of the Russian Academy of Sciences is the leading competency center in the field of system programming in Russia. ISP RAS experts create high-end technologies that allow the Institute to compete with the R&D centers of international IT corporations and the world's top scientific research organizations in diverse areas of system programming — source code analysis, verification, data analysis, operating systems, etc.

The Institute's success is based on an ecosystem that supports complete development chain, from generation of ideas and basic research through technologies and products ready for transfer to customers to actual deployment, and allows training of highly qualified IT experts. Thanks to the founder of the Institute Victor Ivannikov, ISP RAS has preserved the scientific school, which began its formation during the Soviet era, and has been adapted to the modern realities. Nowadays, it shows its viability and effectiveness in the conditions of high mobility of scientists, ideas, and global competition.

The key mechanism for retaining advanced positions is focusing on science-intensive innovations which are based on long-term research projects and sustained partnership. Technologies are developed in close integration with the industry, transferred to and used in leading Russian and international companies. Among the long-term partners of the Institute are Samsung, Huawei, Dell EMC, HPE, Intel, Nvidia, Rogue Wave, Linux Foundation. Many of them have created joint laboratories with the Institute. ISP RAS performs joint projects with leading research and university centers, such as Cambridge (UK), Carnegie Mellon (USA), INRIA (France), University of Passau (Germany) and others.

The Institute provides its own postgraduate program as well as hosts system programming departments of the leading Russian Universities: Moscow State University, Moscow Institute for Physics and Technology, and Higher School of Economics. From the very beginning of their educational programs, students and postgraduates are involved in real-life scientific projects. By the graduation time many students have scientific publications and become skilled engineers in the field of system programming.

The catalogue opens with the description of the ISP RAS business model, which details the structure of the Institute's ecosystem, reveals our approaches to education. The main part of the catalogue provides description of the ISP RAS technologies that are already in use by the industry or ready for deployment. Due to the course to close cooperation with industrial partners and regarding the specifics of commercial projects, the Institute's products have for a long time been known only to particular professional circles. The technologies described for the first time were demonstrated to the wider public at the 1-st Research and Development Conference of ISP RAS at the end of 2016.

ISP RAS AS AN ECOSYSTEM OF INNOVATIONS

The research activity of ISP RAS is aimed at transferring the results of basic research to industry or to other spheres of use. It means that all the Institute's activities focus on ensuring that the technologies, software products, methods for solving problems of system programming created at the Institute meet modern requirements and are maximally ready for adoption by the industry.

The Institute's business model consists of three closely related activities, which together provide a synergistic effect:

- project-oriented basic and applied research in the field of system programming (under contracts with Russian and international companies, the governmental programs, scientific foundations), aimed primarily at creating new technologies;
- innovations - projects to transfer the results of advanced research to the industrial partner companies. An innovative product is impossible without the feedback from the industry;
- education - training students and postgraduates on the basis of modern technologies developed and used in ISP RAS works, involving students in research and industrial projects of the Institute.

The model is well known and used both in research labs of top-ranked universities (Stanford, MIT, Berkeley, Carnegie Mellon), and in laboratories of industrial giants (for example, IBM and Intel), as well as in public research centers such as INRIA (France), Fraunhofer (Germany), and others. This efficiently implemented model helps bridge the gap between science and industry, as well as to train highly qualified researchers and engineers capable of creating and implementing new technologies.

BASIC RESEARCH

The Institute’s model presents all the elements of the chain from generation of ideas and basic research to technologies and products ready for deployment. At the same time, basic research, conducting applied research and prototyping are necessary elements of the activity moving in line with the newest technologies. Basic research is also the source of ideas for new joint projects with partners and customers. All the Institute’s innovations are research-based.
ISP RAS conducts a large number of scientific and educational programs in close cooperation with leading research and university centers in Russia and abroad, that provides high level of research results.

RESEARCH COOPERATION

Joint laboratories are one of the forms used at ISP RAS for organizing long-term cooperation. Having been provided with sustainable funding, they allow flexible planning of the resources available, as well as building competencies in emerging areas of system programming. Also, they help organize training of young specialists with competencies in areas that are of interest to the partners.

Currently, there are a number of joint laboratories at the Institute with international companies: the lab with Samsung (South Korea) is aimed at compiler technologies in the context of mobile platforms (ARM, Android, Tizen); the lab with Rogue Wave (USA) is for technologies of static code analysis and security vulnerabilities detection; Dell (USA) - Big data analysis and processing; Synchro (UK) - comprehensive planning and nD-modeling of large-scale projects;
CUDA Research Center with NVIDIA - parallel computing technologies for modern and promising heterogeneous systems of ultra-high performance. Joint projects are being conducted with leading research and university centers from EU and USA.

A laboratory for solving continuum mechanics problems, based on the technological service-oriented cloud platform FANLIGHT, has been created and evolves rapidly. System and applied software engineers work side-by-side in the laboratory to efficiently implement research projects for the benefit of the industrial enterprises. Such a laboratory ensures the integration of science, education and industry at the modern technological level.

Intellectual property

Using its own technologies and existing back-up for basic research, ISP RAS has created a model that allows it to reserve all the intellectual property rights or transfer them within the framework of special agreements (for example, with the Free Software Foundation) to the community of free software developers. Taking into account the specifics of the Institute's business model, an original license has been introduced. Rather than obtaining royalties, its goal is direct investment by the customer into further research aimed at the development of the technology.

Non-exclusive user rights are given to the customer, whereas all exclusive rights are reserved by the Institute. In specific situations, the decision on intellectual property rights management is taken individually, regarding the prospects for the long-term development of the research direction and the staff of the Institute as a whole. As an example of such an exemption can serve a contract with the Foundation for Advanced Studies (FPI), under which all rights must be transferred to the customer (FPI), as well the customer will be given the non-exclusive patent rights that belong to ISP RAS and are supposed to be used in this project.

Free software

One of the most important components of the created ecosystem is the wide use of free software (FS) — you can hardly imagine modern system programming without it. The Institute considers FS as:

- a tool that provides legitimate free access to a wide variety of modern technologies, including ready-to-use software products, technologies and open standards;
- an opportunity to interact with the global market of products and services, which allows innovative development instead of outsourcing;
- a powerful educational resource: the environment and infrastructure of international FS-projects can be used to train highly qualified experts.

Scientific activity implies openness of a research result and «visibility» of the author of this result, which often comes into conflict with the corporate policy of IT companies. For ISP RAS, openness of research results (in particular, active use of the open source code model) is both working incentive and an instrument for promoting the Institute and transferring the technologies being developed. Openness leads to the fact that young researchers, even though working in a large team, are «visible» in the international community of IT experts. This contribution bolsters their reputation and assets which are enhanced by the Institute.

EDUCATION

Educational activities are the cornerstone of the ISP RAS innovation ecosystem .Integration of ISP RAS with leading universities: departments at Moscow State University, Moscow Institute of Physics and Technology and Higher School of Economics. 50-60 third-year students come annually to the Institute to work on their B.Sc. thesis. In the first year of training at the ISP RAS, they listen to lectures of experts, attend special seminars and get acquainted with the topics on the Institute's research directions. In the second year, the students already participate in the real projects of the Institute. By the graduation time, many students will have scientific publications and will have already been real specialists in their fields of system programming research.

Postgraduate training program. Studying in the ISP RAS graduate school is both accumulation of practical experience and study of new technologies. Moreover, graduate students are actively involved in the teaching process. They conduct seminars and practical classes with students, mentor their course and diploma papers. Having accumulated such experience, a graduate of the degree program heads a small research group, as a rule.

For the decade the number of employees involved in educational activities has increased more than twice. More than a dozen new courses have been developed while the existing ones have been significantly revised. Educational activities are prioritized, so in order to boost effectiveness of training courses, the Institute takes the strategy of involving in the teaching process approximately the same number of the official staff and additional employees.

Starting from the first year of training at ISP RAS, students receive a scholarship; from the second year, a salary that is now comparable to salaries in high-tech IT companies.

Involvement of students and postgraduates in real research projects is a very serious motivation that attracts young professionals to the ecosystem created by the Institute. Research considers a scientific problem or a task to be solved as a real challenge. It requires not only possession of the most advanced technologies, but as well solving such problems beyond the boundaries of known methods.

ISP RAS has created two external laboratories: one is based at the Yerevan State University (Armenia) and the other is at Yaroslav-the-Wise Novgorod State University (Russia). The staff of the external labs participates in industrial and research projects with ISP RAS, forming a distributed development and research environment, working on joint publications and conference talks, and delivering modern courses at their universities. Young specialists of the laboratories do a full-time internship at ISP RAS, including semester-long training for the preparation of their final theses.

ISP RAS IN FIGURES

With a total increase in funding from 105 million rubles in 2005 to 674 million rubles in 2016, the share of contract work is about 70-80% of the total, consistently. And in the total volume of contract work, the share of work with Russian organizations has increased from less than 3% in 2005 to a half in 2016. At the same time, the average salary of researchers grew from 26,000 rubles in 2005 to 180 thousand rubles in 2016.

The number of staff researchers increased more than twice from 2005 to 2016, including 10 Dr.Sc. and 34 PhD. Given that, the share of young scientists in 2005 was just over 50%, and at the end of 2016, researchers under the age of 40 accounted for about 80% of the total number.

Employees of the Institute took part in more than 300 leading Russian and foreign conferences. Over a thousand scientific articles and ten monographs have been published.

Junior staff members regularly receive scholarships from the Government and the President of the Russian Federation. Also, two young researchers have been awarded medals of the RAS and prizes. Two staff members were awarded State Prizes of the Russian Federation.

ISP RAS publishes a journal named “Programming and Computer Software” that is included in the international science citation indexes Scopus and Web of Science (Core Collection).

QUICK FACTS

The ecosystem created in ISP RAS is based on advanced approaches to the organization of research and development, including the widespread use of open source software and management of intellectual property. It has a high level of adaptability and dynamics, which enables adequate respond to the emerging technological and organizational challenges. It greatly expands the horizons of planning and allows building relationships with customers and partners on a long-term basis, significantly reducing the risks for both parties. This is important, since collaboration with organizations of the real sector of the economy is the key component of the Institute’s ecosystem.

It is impossible to achieve and sustainably maintain high technological level of research and development without such cooperation.

Since the establishment the Institute has accumulated considerable experience and has obtained world-class results in a number of advanced areas of system programming. The Institute has created its own technologies and instruments, implemented dozens of industrial projects, obtained a number of patens in the ICT domain, trained hundreds of students and postgraduates; staff members of the Institute have presented more than 50 Dr.Sc. and PhD dissertations and brought out hundreds of publications.

The work of ISP RAS as an effective scientific and production center in the field of system programming has gained worldwide recognition. In particular, it is expressed through the inclusion of ISP RAS scientists in the program committees of many international conferences, their invitations as keynote speakers and participation in expert groups of international consortia. ISP RAS has created two competence centers for specialized areas - the Linux OS Verification Center and the Center for Parallel and Distributed Computing, which have been successfully integrated into the Russian and world community and into all the activities on the ISP RAS model.

TECHNOLOGIES

SVACE

Smart detection of errors
and vulnerabilities

STATIC ANALYZER

Svace is an essential tool of the secure software development life cycle, the main static analyzer that is used in Samsung Corp. It detects more than 50 critical error types. Svace supports C, C++, C#, and Java. Svace is registered in the National Software Unified Register, which is kept by the Ministry of Digital Development.

WHY SVACE?

1 Maximum flexibility and adaptability for Russian customers

Our team is based in Moscow and provides fast customer tailored deployment and short support turnaround. Svace has no Russian competitors and offers the maximum level of convenience and efficiency for Russian users:

- Accelerated customization (configuring existing detectors as well as writing individual ones available exclusively to this customer; creating specific user interfaces);
- Ultra fast adaptation to new environments and tools (adding new compilers within 1-2 weeks, in complex cases   up to 2 months);
- Continuous training of customer’s developers, regular interaction with the customer, providing technical product improvements with solving the customer’s new tasks during the whole secure software development lifecycle;
- Flexible licensing terms tailored to the customer’s needs (in particular, the possibility to obtain the product source code);
- Full compatibility with regulatory documents and requirements of regulators (FSTEC of the Russian Federation);

2 Quality at the level of international competitors

Svace is a constantly evolving innovative product based on years of research. It combines the key qualities of foreign competitors (Coverity Static Analysis, HP Fortify, RogueWave Klocwork Static Analysis) with the unique open industrial compilers usage to provide the maximal support level for new programming language standards.

Svace is defined by:

- High-quality deep analysis:
  - an accurate representation of the source code (due to integration with any build system);
  - full path coverage taking into account function calling contexts for searching complex errors;
  - high percentage of true positives (60-90%).
- Scalability and high speed:
  - parallel analysis using up to 64 processor cores;
  - ability to analyze software with the code size of tens of millions of lines (analysis of Android 6 consisting of 8 million lines takes 5-6 hours);
  - supporting incremental system analysis in addition to the full analysis mode (implies a quick re-analysis of the recently modified source code).
- Convenient warnings viewing interface:
  - detailed error description with code navigation;
  - review interface for marking true and false positives;
  - analysis results migration between runs with hiding any issues previously marked as false positives.

WHAT IS SVACE TARGET AUDIENCE?

- Companies aimed at software development with a special focus on high reliability and security;
- Companies that need certification of developed software;
- Certification laboratories.

SVACE AND SAMSUNG

Svace is the main static analyzer used in Samsung Corp. since 2015. It is used to check the company’s own software based on Android OS as well as the Tizen OS source code. Tizen is used in smartphones, infotainment systems and Samsung home appliances. Since 2017, Svace checks all changes submitted for review and inclusion in the Tizen OS.

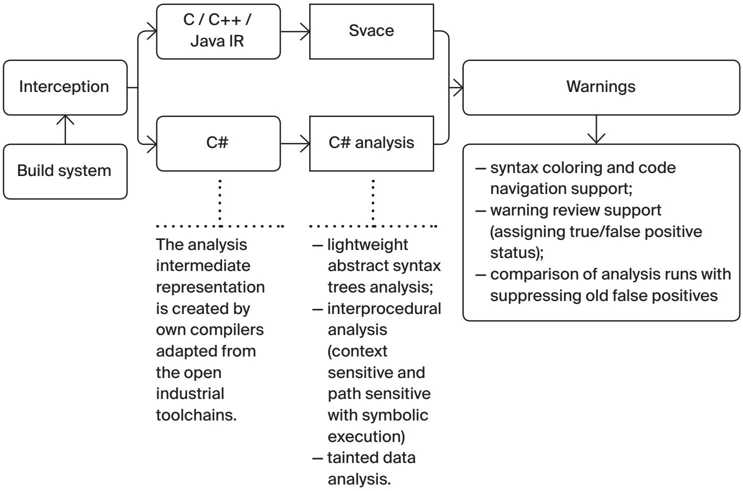
SUPPORTED PLATFORMS AND ARCHITECTURES

- Host platforms for the analyzer: Linux kernel based OS (version 2.6 and later), Windows XP and later.
- Target architectures of the analyzed code: Intel x86/x86-64, ARM, ARM64, MIPS, MIPS64, Power PC, Hexagon.

SUPPORTED COMPILERS

- For C/C++: GCC (GNU Compiler Collection), Clang (LLVM compiler), Microsoft Visual C++ Compiler, RealView/ARM Compilation Tools (ARMCC), Intel C++ Compiler, Wind River Diab Compiler, NEC/Renesas CA850, CC78K0(R) C Compilers, C/C++ Compiler for the Renesas M16C Series and R8C Family, Panasonic MN10300 Series C Compiler, C compiler for Toshiba TLCS-870 Family, Samsung CalmSHINE16 Compilation Tools, Texas Instruments TMS320C6* Optimizing Compiler.
- For C#: Roslyn, Mono.
- For Java: OpenJDK Javac Compiler, Eclipse ECJ compiler, Jack Compiler for Android.

SVACE ARCHITECTURE



The analysis intermediate representation is created by own compilers adapted from the open industrial toolchains.

- lightweight abstract syntax trees analysis;
- interprocedural analysis (context sensitive and path sensitive with symbolic execution)
- tainted data analysis.

- syntax coloring and code navigation support;
- warning review support (assigning true/false positive status);
- comparison of analysis runs with suppressing old false positives

BINSIDE.
STATIC BINARY CODE ANALYSIS TOOL

THE IMPORTANCE OF BINARY CODE ANALYSIS

Software developers often face a problem of incorporating complex computations, data encryption and compression algorithms, and similar common notions into their code. This is typically done by using standard libraries specializing in a group of tasks; these libraries are often distributed in binary code only. On the other hand, software maintenance is gradually becoming more and more important within the development cycle; software maintenance incorporates the task of updating both its code and external libraries. External libraries and auxiliary programs, distributed in binary form, need to conform to quality and security standards.

STATIC BINARY CODE ANALYSIS TOOL

At the Institute for System Programming of the Russian Academy of Sciences we have developed a static binary code analysis tool. This tool includes the following features:

- Support for various processor architectures: x86, x64 (ARM, PowerPC, MIPS to be added);
- Support for various platforms: Linux, Windows;
- Support for various binary code formats: ELF for Linux and PE for Windows;
- Support for binary code analysis without debug information or symbol tables;
- Automatic defect detection: invalid usage of format string functions, buffer overflows, invalid usage of dynamic memory.

PROGRAM ANALYSIS

The analysis tool transforms executable and library binary code into the specialized architecture independent internal representation used to create control flow graphs and call graphs; these graphs are used to perform context-sensitive intra-procedural data flow analysis in order to identify potential runtime defects and vulnerabilities. The context-sensitive analysis core automatically generates function specifications and propagates them through function call code points.

DEFECT DETECTION

We currently provide automatic checkers that identify problems with format string functions, potential buffer overflow defects and invalid usage of dynamic memory.

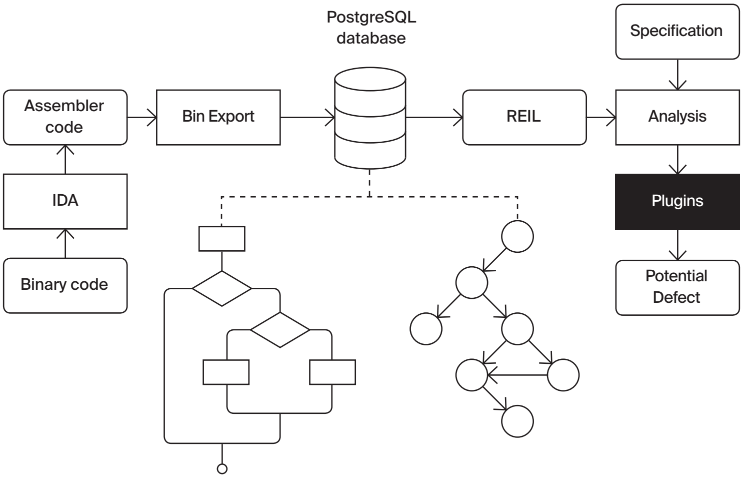
EXTENSIBILITY FEATURES

Our analysis tool provides an API for accessing internal code representation and models and can be used to design new checkers.

INTERNAL INFRASTRUCTURE

Our analysis tool employs IDA Pro — a de facto standard in the field of program disassembly and reverse engineering — and additional tools (Google (Zynamics) BinNavi and BinExport) modified to our needs; these tools transform program binary code into REIL — an architecture-independent intermediate representation language. We are extending these tools in order to improve the efficiency of intraprocedural analysis, abstract interpretation, defect detection, tainted data flow analysis, PDG (Program Dependence Graph) generation and other methods. Certain extensions (e.g. x64 support for REIL transformation) were successfully released into the community.

OPERATIONAL SCHEME



ANXIETY DYNAMIC ANALYZER

Combined approach to finding errors

Anxiety is a framework for detecting errors and potentially dangerous cases in the process of development, acceptance testing and operating the software. It is based on the dynamic symbolic execution, which allows automatically generate input data with no source code or debugging information available.

WHY ANXIETY?

1 Combination of essential functions

Anxiety's special feature is the combined approach to dynamic analysis, which involves the integration with static analyzers and fuzzing tools. The successful combination of technologies allows Anxiety to solve the same tasks as the leading global analogues (CA Veracode Dynamic Analysis, Synopsys Dynamic Application Security Testing and Rogue Wave CodeDynamics)

Anxiety provides following features:

- High level of analysis performance due to support of distributed and parallel modes of operation, integration with a fuzzer, as well as filters for input data stream and analyzed functions;
- Creation of analysis tools based on the dynamic symbolic execution method;
- Integration with static analyzers of source or machine code for the implementation of directed analysis, which allows testing of the components for the target program selectively. Verification of defects previously detected by static analysis (in particular: division by zero, dereferencing of a null pointer, infinite looping, violation of user asserts and etc.);
- Integration with tools for programs randomized testing (fuzzing) to increase its performance. Integration solves problems of randomized testing when it is faced with passing conditional transitions that depend on comparison with constants. Fuzzing usage makes it possible to achieve the program's code coverage with input data sets much faster than in case of dynamic symbolic execution;
- Modular infrastructure (tracer, checker and input data generator) allowing to adopt system components to analysis needs and expand its functionality;
- Support of various sources for external program data (files, network sockets, environment variables, standard input flow).

2	Convenience for Russian customers	Anxiety is a cutting the edge development of the Institute for System Programming of the Russian Academy of Sciences based on the results of long-term of research and intended for industrial usage. Flexible basic environment with the ability to fully adapt to the needs of the customer. The benefits are: — Implementation of specific tasks of program analysis based on dynamic symbolic execution (in particular, determining the reachability of a certain function or operation in a program); — Ability to receive the alienable product; — Ability to be used for the implementation of interim requirements of GOST R 56939-2016 (in case if software certification is needed for deploying in Russia)

WHO IS ANXIETY INTENDED FOR?

- Companies aimed at software development with a special focus on high reliability and security;
- Companies responsible for software audit or certification.

WHERE IS ANXIETY USED?

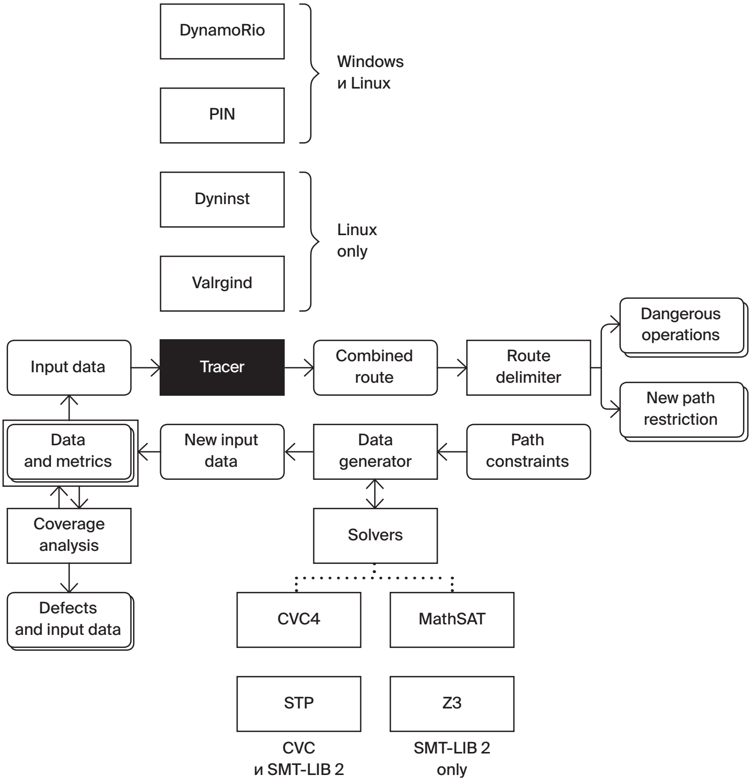
The Anxiety tool is used for testing programs included into the Astra Linux OS packages.

SUPPORTED ENVIRONMENTS AND TOOLS

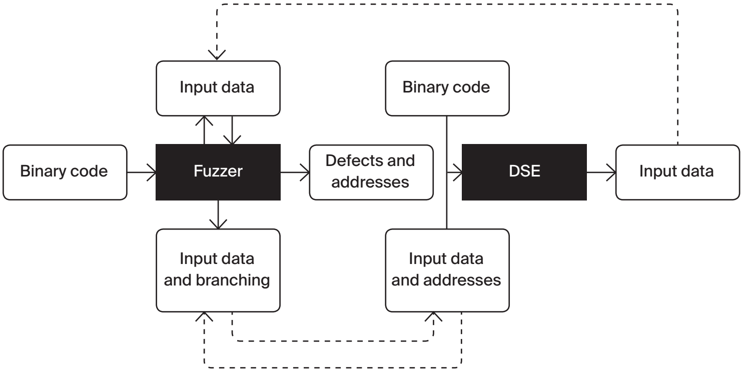
Anxiety supports an analysis in Windows OS (XP version and higher) and Debian Linux OS, as well as the operation with various types of SMT solvers (STP, Z3, MathSAT, etc.). It is based on DynamoRIO dynamic instrumentation environments (instruction flow is processed by the Triton framework, and supports Windows OS programs analysis) and Valgrind dynamic binary instrumentation, which used for trace interception and automatic basic block coverage mechanism.

OPERATIONAL SCHEME

Symbol dynamic analysis



Fuzzing



BINARY CODE ANALYSIS PLATFORM BASED ON QEMU EMULATOR

QEMU is a full-system multi-target open source emulator. It is widely used for software cross-development. Many large companies (e.g., Google, Samsung, Oracle) prototype and emulate their hardware platforms and peripheral devices on QEMU.

Open-source code allows extending Qemu features to use Qemu for:

- creating new virtual platforms,
- prototyping peripheral device models,
- debugging OS kernel code, firmware code, drivers for emulated devices,
- malware analysis,
- recording virtual machine execution for later replay and analysis.

REMOTE DEBUGGING IN THE EMULATOR

QEMU supports remote debugging of virtual machine through the GDB-compatible interface. Debugging service works within the emulator and does not affect virtual machine behavior.

GDB (open source debugger) can connect to the emulator via network sockets and inspect processor registers, memory cells, call stack, and so on. One can debug either application or kernel code in the virtual machine. Popular binary analysis tools and IDE such as IDA and Eclipse can also connect to QEMU for debugging and analysis of the virtual machine, because they support GDB-compatible remote debugging interface.

VIRTUAL MACHINE EXECUTION RECORDING AND REPLAY

Debugging usually needs to trace from the failure to the line of code where an error actually appeared. It implies moving “back in time”. To restore past program state one has to re-run it and try to find failure source. This operation is usually performed multiple times, moving backward step-by-step.

Debugging is significantly more difficult if its manifestation is unstable: it is affected by “random” factors, such as multithreaded execution, hardware behavior, user interactions with graphical interface and so on. Deterministic replay provides stable reproduction of a program (or virtual machine) run, and thus facilitate debugging.

Deterministic replay reconstructs program execution using previously recorded input data. The first program run is used to record these inputs into the log. Then all following runs will reconstruct the same behavior, because the program uses only recorded inputs. Deterministic replay reconstructs the sequence of program (or virtual machine) states including CPU registers, memory cells, peripheral devices’ state, and hard disk contents. Replay proceeds between these states executing CPU instructions and passing previously recorded inputs to the program. These inputs include user input, network packets, serial and USB communications.

Full-system replay may be used for analysis of user-level applications, system kernels, firmwares, and multi-threaded programs. Every guest operating system supported by the emulator may be recorded and replayed.

Every replay run produces equivalent executions (the same sequence of the instructions and hardware states) and therefore may be used for convenient debugging of volatile bugs. Debugger and other analysis tools do not alter program execution, because they work outside of the guest system. Deterministic replay in QEMU is created by ISP RAS. QEMU allows recording and replaying virtual machine executions for x86, ARM, and MIPS platforms.

REVERSE DEBUGGING

Reverse debugging may be used to inspect past states of the program. Developer starts debugging from the point where an error manifests itself or exception occurs. Then he tries to determine the reasons of such behavior. Usually the failure is caused by some operations performed in the past.

Reverse debugging does not require restarting of the program, because it assumes assumes faster “rewind” to the past. GDB interface includes “reverse step” and “reverse continue” commands. Implementation of these operations in QEMU uses deterministic replay and virtual machine snapshots for faster recovering of the past states. These patches later will be included into QEMU mainline.

GUEST SYSTEM ANALYSIS

Virtual machine debugging requires information about programs and modules location in memory. We have developed introspection mechanism which gets such information from virtual machines with Windows or Linux inside. Introspection can be used for retrieving:

- instruction execution sequence,
- memory access sequence,
- executing system calls,
- created processes,
- loaded modules,
- file accesses

NEW PLATFORM AND PERIPHERAL DEVICES EMULATION

Emulating new devices and platforms in QEMU requires a complete set of documentation describing the instruction set architecture of the processor, memory map and peripherals. Every new peripheral device must be provided with its own documentation.

Development of a new platform in the Qemu emulator from scratch requires implementation of:

- new virtual CPU and translator for its instructions into intermediate representation,
- virtual memory management unit (MMU),
- virtual peripheral devices,
- new platform which integrates all of the above,
- extension of QEMU interfaces for new devices connected to the real world.

Even when QEMU already includes implementations of virtual CPU, MMU, and peripheral devices, all of these parts need to be interconnected with virtual system buses into one virtual platform.

In case of lack of documentation or its incompleteness, virtual platform debugging becomes very difficult. Information about the platform may be extracted only from available binary code for the existing devices. Then code execution failures provide information about virtual hardware implementation flaws. Emulator development requires more efforts in this case, because binary code analysis is used to recover expected behavior of the virtual device.

We provide semi-automatic scripts on Python to simplify new virtual platform development. It provides declarative API for configuration description and graphical interface for making this configuring simpler.

OBFUSCATOR

Obfuscator is a set of technologies to prevent mass exploitation of vulnerabilities resulting from errors or bookmarks. If the hacker is able to attack one of the devices with the common software, the rest will remain protected by changes made to the code.

WHY OBFUSCATOR?

1 Fast adaptation in accordance with customer needs

Obfuscator is based on years of code obfuscating transformations research. Due to the presence of all the necessary competencies, our team is able to create an individual industrial solution for any customer:

— Fast adaptation of the technology to a specific compiler or binary code toolkit (adaptation takes 2 months on average and up to six month in difficult cases);
— The ability to get a completely alienable product;
— Free testing and demonstration (using the customer's program or other);
— Continuous adjustment of transformations and technological adaptation of the product in accordance with new tasks and challenges.

2 The optimal combination of the necessary features

Obfuscator protects the system from mass exploitation of vulnerabilities using various methods of code diversification and allows compiling the code of full OS.

Obfuscator is defined by:
— Fine-tuning of balance between the degree of obfuscation and the level of performance (when used to protect against reverse analysis). The minimum deceleration is 1.2 times, the maximum is 8 times;
— Full automation (no special arrangement of the program source code or additional efforts from the customer's build engineers are required);
— Usage of a set of GCC open compilers, which allows compiling OS correctly;
— Two methods of diversification:
  - Dynamic code diversification at program startup. It is used when the customer needs the same code on all devices (for example, due to mandatory certification). This method allows you to transfer up to 98% of the code with a slight increase in its volume and performance degradation by about 1.5%. Obfuscator advantages over the similar products are the following:
    - Obfuscating down to the functions (as opposed to

ASLR and Pagerando technologies that obfuscate only large blocks of code);

- Obfuscating functions throughout the system, except for the kernel, and the absence of potential conflict with antiviruses (advantages over similar Selfrando technolovgy developed for the Tor Browser);
  - Static code diversification. During each compilation, depending on the specified key, a new executable file is obtained. The advantages of this method are the following:
    - the amount of binary code is not increased (which is particularly important for the Internet of things);
    - deterioration of performance tends to zero;
    - thanks to the operations being done inside the compiler, but not in the program linker post factum, an extended set of diversifying transformations can be applied and more flexibly customized.
- Conflict-free combination with other software protection tools (including the ASLR system mechanism).

WHO IS OBFUSCATOR INTENDED FOR?

- Developers of specialized OS;
- Application software developers.

WHERE IS OBFUSCATOR USED?

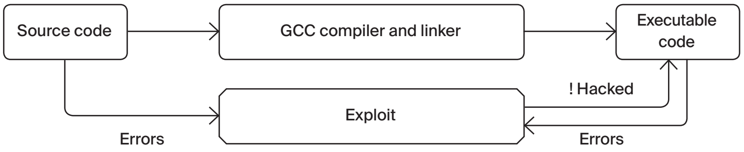
ISP Obfuscator is implemented in the Zirkon OS, which is used by the Ministry of Foreign Affairs and the Border Guard Service of the Federal Security Service of Russia.

SYSTEM REQUIREMENTS

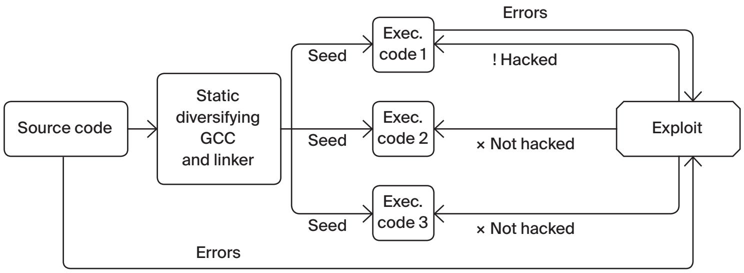
Obfuscator is a universal product that can be adapted to any system requirements. The main version is currently running on a Linux-based OS (starting with version 2.6) with the Intel x86 / x86-64 architecture support.

OPERATIONAL SCHEME

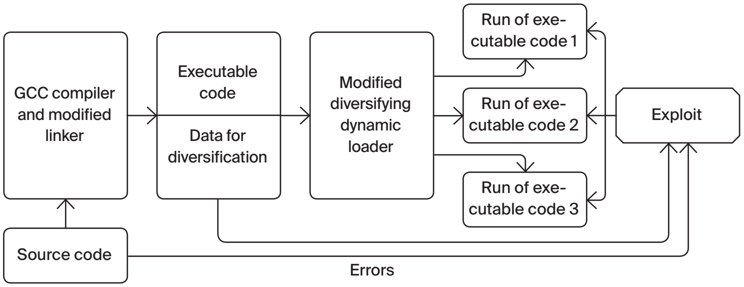
Standard Compilation



Static Diversification



Dynamic Diversification



PROTOSPHERE NETWORK TRAFFIC ANALYZER

Analyzes traffic, detects anomalies

Protosphere is a system of deep packet inspection (DPI). It is the part of intrusions and information leaks protection. Detects inconsistencies between protocol specification and specific implementation. Allows you to quickly add support for new (including closed) protocols due to the flexibility of the internal representation.

WHY PROTOSPHERE?

1 Optimal combination of necessary functions

Protosphere is an innovative system based on the research of network traffic analysis technologies. Combines key features of foreign competitors (Wireshark, Microsoft Network Monitor) with a universal internal presentation that allows you to quickly expand the analysis capabilities.

Protosphere is defined by:

- Optimal system core capabilities:
  - universal model of data representation in the process of network traffic analysis;
  - processing of data containing distortions, losses, rearrangements and duplication of packets, as well as asymmetric traffic;
  - support of compressed and encrypted data analysis;
  - support of arbitrary configuration tunnels.
- Support of all stages of network traffic research through synchronized visualization tools:
  - localization of one or more investigated network connections on the graph of network interactions and the network flows tree;
  - providing details for the selected connections on the flow chart;
  - visual representation of the analysis results on the analysis tree;
  - diagnosis of inconsistencies between the protocol specification and the actual traffic in the malfunction diagnosis log;
  - extracting and analyzing data, including the application layer, by shared use of data content windows, a list of fragments and a list of objects.
- Fast expansion of the supported protocols list:

		– access to the analysis results API; – localization of parsing errors; – the ability to debug the module under development on live traffic, which allows significant acceleration of new protocols introduction. — Supports two modes of operation: online and offline mode. Due to the unified code base, the implementation and verification of new protocols is accelerated; — An advanced graphical interface that allows you to choose the most convenient way to present the results of the analysis.
2	Flexibility and adaptability to customer needs	— Accelerated customization due to the flexibility of the internal presentation (support of new protocols, extraction of new data types, setting up the format for analysis results); — Adaptation for network channel and available computing resources (flexible configuration system allows you to find a balance between detail and accuracy of analysis and consumed resources); — Ability for the customer to obtain an alienated product.

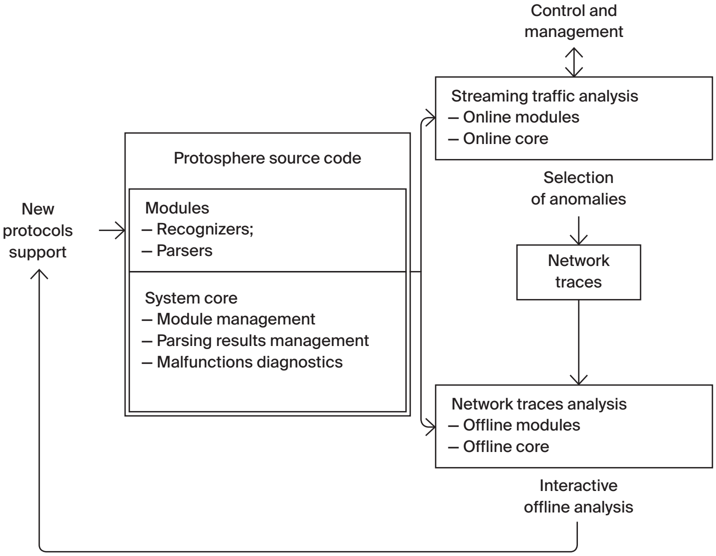
WHO IS THE PROTOSPHERE SYSTEM INTENDED FOR?

- Companies involved in testing network protocols implementations (including embedded operating systems and network equipment);
- Companies-developers of network security tools (firewalls, intrusion detection and prevention systems);
- Companies manufacturing equipment that needs an increased level of safety due to mandatory certification.

SUPPORTED PLATFORMS AND ARCHITECTURES

- Architecture: Intel x86-64.
- Platforms: Windows OS, Linux kernel based OS.

OPERATIONAL SCHEME



KLEVER:

Omits no mistakes, guarantees safety

THE TECHNOLOGY OF STATIC VERIFICATION OF GNU C PROGRAMS

Klever is a static verification system that uses advanced tools to thoroughly check the security, reliability and performance of software systems developed in the GNU C language. In particular, it is used to verify the real-time OS.

WHY KLEVER?

1 High accuracy and advanced analysis capabilities

Klever is a research-based technology intended for industrial use. It allows using highly accurate formal methods for verifying large software systems. It solves the same tasks as its global analogues (for instance Microsoft SDV), while being different in the ability to verify any complex software, but not just a narrow class of programs (drivers or embedded software). The technology is available in open access (forge.ispras.ru/projects/klever).

Klever is defined by:

— High-precision sound analysis of any complex software (identification of all possible errors of the desired types);

— Checking various requirements for the program (checking the rules of safe C programming and correct usage of the interface specific to the program being checked);

— Scalability of formal methods (modular static verification allows you to scale the use of appropriate tools and analyze projects that contain thousands of lines of code in the GNU C language);

— Support of verification of different versions of the program (no need to modify the source code of the program being checked, which makes it easy to verify various versions and verify the correction of errors immediately after their detection and elimination);

— Using the incremental process of refining the results of verification (a constantly refined set of specifications is used to control the decomposition and generation of models of the environment of program fragments)

2	High level of adaptability and convenience	— Adaptation of technology to the needs of the customer. Timely expansion of the list of detectable errors. Development of specifications set for formalizing the program-specific requirements, as well as the environment modeling specifications and, in some cases, plug-ins; — Convenient multi-user web interface for performing static verification, as well as storing, analyzing and comparing results; — Ability for the customer to obtain an alienated product (after preparation, adaptation, development of specifications and search for errors). Training of customer's developers;

IMPLEMENTATION EXPERIENCE

The Klever technology was developed as part of the Linux Verification Center (http://linuxtesting.org/) supported by the Linux Foundation and organized on the basis of the Institute for System Programming of the Russian Academy of Sciences. Klever is currenlty used to verify various operating systems.

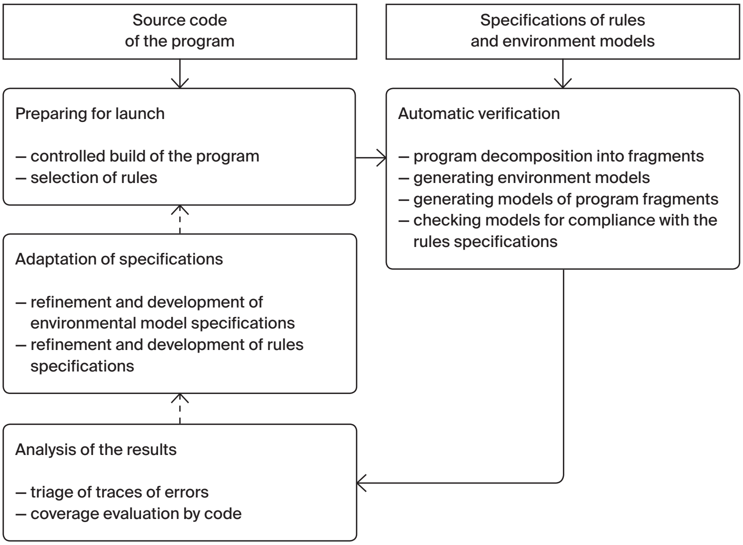
In the process of verifying the device drivers and subsystems of the Linux OS, the following results were achieved:

- More than 300 developers confirmed errors were detected: buffer overrun errors, null pointer dereference, use of uninitialized memory, repeated or incorrect memory de-allocation, race conditions and interlocks, leaks of specific Linux kernel resources, incorrect function calls depending on the context, incorrect initialization of specific data structures of the Linux kernel;
- 50% coverage of the device drivers and kernel subsystems achieved. In order to look for these errors when executing various scenarios of interaction between Linux kernel drivers and their environment, Klever built a fairly accurate environment model (more than 20 of the most widely used driver interfaces, such as interrupt handlers and timers, USB and PCI devices' interfaces, network and character interfaces);

WHO IS KLEVER INTENDED FOR?

- Companies aimed at software development with a special focus on high reliability and security;
- Companies that need certification of developed software;
- Certification laboratories.

OPERATIONAL SCHEME


Source code of the program
Specifications of rules and environment models
Preparing for launch
— controlled build of the program
— selection of rules
Automatic verification
— program decomposition into fragments
— generating environment models
— generating models of program fragments
— checking models for compliance with the rules specifications
Adaptation of specifications
— refinement and development of environmental model specifications
— refinement and development of rules specifications
Analysis of the results
— triage of traces of errors
— coverage evaluation by code

MICROTESK. TEST PROGRAMS GENERATOR

Verifies microprocessors

MicroTESK is a reconfigurable and expandable test program generation environment for functional microprocessors verification. It allows automatically constructing test program generators for target microprocessor architectures based on their formal specifications. MicroTESK is applicable for a wide range of architectures (RISC, CISC, VLIW, DSP).

WHY MICROTESK?

1 Sophisticated and promising concept

MicroTESK is a technology stack for industrial use, which includes the basic modeling environment (it builds models of microprocessors based on formal specifications) and the generation environment (it builds test programs based on templates). Based on the tasks solved, it is close to its global analogues (Genesys Pro and RAVEN), however, it differs from them in increased productivity and usability, as well as distribution under the open source license.

It is free for access on the Institute for System Programming of the Russian Academy of Sciences website: https://forge.ispras.ru/projects/microtesk. In addition, a description of the technology is available at http://www.microtesk.org/.

MicroTESK is defined by:

- Using formal specifications as sources of knowledge about the configuration of a verifiable microprocessor:
  - specifications of nML architectures (registers, memory and addressing modes, instruction logic, text/binary instruction format);
  - additional specifications of the memory subsystem on mmuSL (properties of memory buffers (TLB, L1 and L2), address translation logic and read and write operations logic);
  - the potential to move to a formal verification and to the generation of a set of tools for the microprocessor under development (disassembler, emulator, etc.);
- Generation of test programs based on object-oriented test patterns:
  - test patterns in Ruby language (due to which the patterns are graphic and easy to support);
  - the possibility of simultaneous use of different generating techniques for sets of instructions and test data (random generation, combinatorial generation, generation based on the resolution of restrictions, etc.);

  - scalability of the generation environment (the ability to develop complex templates at low cost due to reuse).
- Wide range of supported microprocessor architectures:
  - support of features of various classes of architectures at the level of generators design environment (RISC, CISC, VLIW, DSP);
  - MicroTESK-based test program generators were developed for such architectures as ARM, MIPS, PowerPC, RISC-V;
  - Multi-core architecture of target microprocessor is supported.

2 Maximum convenience for the customer

- Timely setup of the environment for new architectures with minimal costs and automatic extraction of information about test situations (due to formal specifications);
- Convenient language for developing test patterns that allows quickly describing complex verification scenarios;
- Possibility to integrate a wide range of different generation methods (random, combinatorial, based on the resolution of restrictions, etc.);
- Timely local technical support and training of customer’s developers;
- Ability for the customer to receive an alienable product.

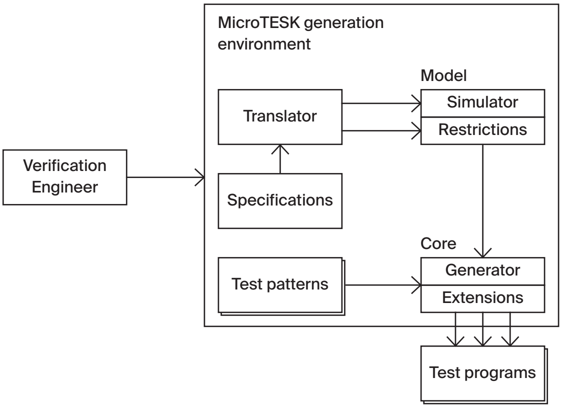
SYSTEM REQUIREMENTS

Windows OS or the GNU\Linux kernel based OS, Java Runtime Environment version 8.

IMPLEMENTATION EXPERIENCE

MicroTESK has been under development since 2007. It was used in Russian and international projects for the development of modern industrial microprocessors (particularly, in industrial projects for the verification of ARMv8 and MIPS64 microprocessors).

OPERATIONAL SCHEME



RETRASCOPE: HDL-DESCRIPTIONS REVESRE ENGINEERING TOOL

Static analysis of digital hardware descriptions

Retrascope is a tool for reverse engineering and functional verification of digital hardware descriptions. It provides automated tools for extracting and analyzing formal models from source code. The tool supports synthesized subsets of Verilog and VHDL languages.

WHY RETRASCOPE?

1 Combination of the most important qualities

Retrascope is an extensible tool that allows you to develop hybrid verification techniques for HDL descriptions by combining various tools for analyzing formal models.

Retrascope is defined by:

- Extracting formal models from source code and their visualization.
- The following types of models are supported:
  - control flow graph;
  - decision diagram of guarded actions;
  - high-level decision diagram;
  - extended finite state machine.
- Generation of functional tests for hardware modules (random generation, extended finite state machine bypass, bounded model checking);
- Verification of formal models (model checking) for compliance with PSL specifications using external verification tools (NuSMV, nuXmv).

2 Convenience for the customer

- Graphical user interface based on the Eclipse IDE (command line interface is also available);
- Open source code (Apache Licensed Version 2.0);
- Extensibility at the source code level (the ability to add new hardware descriptions or analysis tools);
- Open interaction interfaces (SMT-LIB, SMV languages) allow using various model checking tools and solvers to achieve analysis and verification goals.

WHO IS RETRASCOPE INTENDED FOR?

- Companies aiming to develop digital hardware;
- Research groups in the field of functional verification of digital equipment.

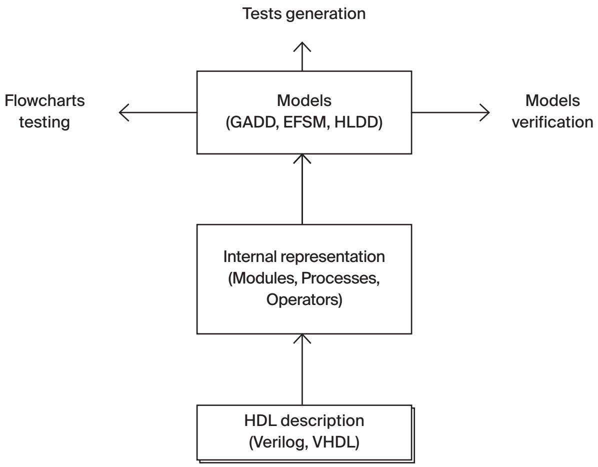
IMPLEMENTATION EXPERIENCE

The tool is at the research prototype stage, development is underway.

SYSTEM REQUIREMENTS

Hardware: IBM-compatible PC.
Software: Windows OS or GNU\Linux kernel based OS, Java Runtime Environment version 8.

OPERATIONAL SCHEME


Tests generation
Flowcharts testing
Models (GADD, EFSM, HLDD)
Models verification
Internal representation (Modules, Processes, Operators)
HDL description (Verilog, VHDL)

ASTRAVER TOOLSET.

DEDUCTIVE VERIFICATION OF LINUX KERNEL MODULES AND SECURITY POLICY MODELS

Software plays a key role in many systems, e.g., safety-, security-, and mission-critical systems. Bugs in such software can lead to catastrophic consequences. As a result development of critical software is regulated by certification standards/guidelines (like DO-178C, ISO/IEC 15408, etc) that require following best practices in development process.

For example, ISO/IEC 15408 “Information technology — Security techniques — Evaluation criteria for IT security” requires including of the following activities:

- formal security policy modelling (ADV_SPM);
- formal verification of internal consistency of a security policy model;
- formal proof that the target system cannot reach an unsecure state;
- development of formal and semi-formal functional specification;
- formal proof of correspondence between the security policy model and the functional specification;
- formal proof of correspondence between different representations of target software like functional specification, design and source code.

ISP RAS has developed methods and tools implementing these activities. The approach has been applied for verification of security module of Astra Linux Special Edition operating system developed by RusBITech.

The approach suggests using two specification languages with corresponding toolsets:

- security policy models and formal functional specifications are specified in Event-B;
- formal specification of critical implementation components is done in ACSL.

EVENT-B AND RODIN

Event-B specification consists of contexts and machines. Contexts contain the static part of a specification: constants, carrier sets, axioms. Machines contain the dynamic part: variables, invariants, events. The current state of a specification is formed by means of variables whose values are constrained by invariants and changed by events.

State invariants allow both ensuring the internal consistency of a machine and formalizing the notion of a “safe” state required by ISO/IEC 15408-3, ADV_SPM.1.2C.

Each event consists of parameters, guards, actions. Guards restrict the values of event parameters and machine variables reducing the number of states in which the event can occur. Actions modify the current state by assigning new values to the variables. Event-B also allows us to decompose specifications using the refinement technique to simplify the development, verification and support processes.

Event-B specifications are developed and verified using the Rodin platform (developed under open source license by ETH Zurich, Systerel, Clearsy, University of Newcastle and University of Southampton) and its plug-ins. Rodin automatically generates proof obligations for each case that requires proof: invariant preservation; well-defined axioms, invariants, guards, actions; refinement correctness. To fully verify the model all generated proof obligations need to be proved. To do this, Rodin allows using various automatic provers (for example, SMT solvers) as well as performing interactive proof.

ACSL AND ASTRAVER TOOLSET

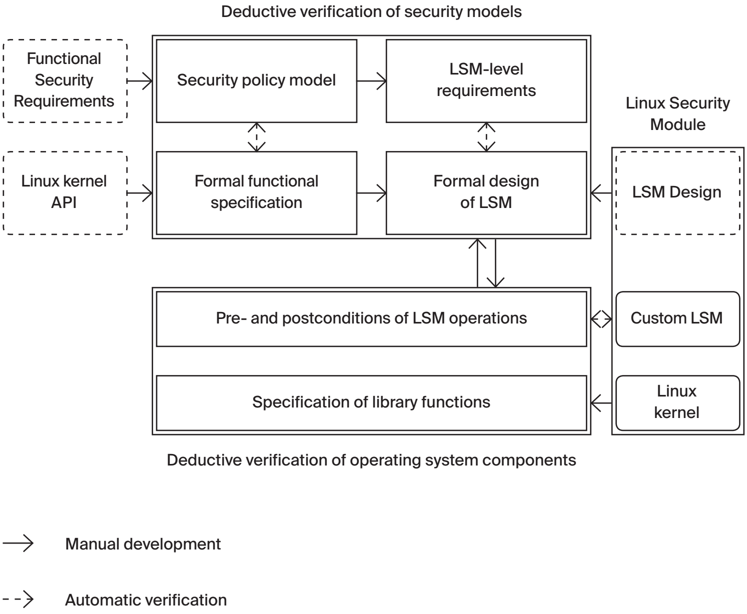
Formal specifications of component interfaces in C language are proposed to be described in a special language called ACSL (ANSI/ISO C Specification Language). ACSL is a C programs behavior specification language that supports contract specifications from the most low-level, such as “this function requires a valid initialized pointer to int as an input”, to high-level, for example: “this function requires a non-empty linked list of int values as an input and returns maximum of these values”. Expressive power of ACSL language is enough to fully specify various functions, and it can also be used to write partial specifications.

Deductive verification is based on the translation of C source code annotated with specifications of checked properties into a set of logical formulae, which general significance is equivalent to the source program being correct in accordance with given properties (if a precondition holds when function is called, then function will finish and its result fulfills its post-condition). These logic formulae, known as verification conditions, can be checked for satisfiability with many different tools: automatic, such as Z3, CVC, Alt-Ergo, Vampire, E-Prover etc., or requiring user participation, for example, interactive theorem provers such as Coq and PVS.

Existing deductive verification tools don’t support all the features of C language used in operating system kernels. As a result Astraver Toolset, a new deductive verification toolset, was developed by ISP RAS. The toolset is based on an open C program verification platform Frama-C (CEA-LIST, France) and deductive verification system Why3 (INRIA, France), and includes the following new features:

- container_of construct support;
- function pointers support;
- expression-level support for bitwise arithmetic operations;
- support for pointer type reinterpretation between integer types, incl. types of different size;
- zero-sized arrays support;
- String literals support;
- Template specifications for standard library memory operations;
- Control flow highlighting for verification conditions in Why3ide.

OPERATIONAL SCHEME


Deductive verification of security models
Functional Security Requirements
Security policy model
LSM-level requirements
Linux Security Module
Linux kernel API
Formal functional specification
Formal design of LSM
LSM Design
Pre- and postconditions of LSM operations
Custom LSM
Specification of library functions
Linux kernel
Deductive verification of operating system components
Manual development
Automatic verification

MASIW.

SOFTWARE TOOLS FOR DEVELOPMENT OF INTEGRATED MODULAR AVIONICS

MASIW — MODULAR AVIONICS SYSTEM INTEGRATOR WORKPLACE

The Modular Avionics System Integrator Workplace framework is intended to automate the design of real-time aviation electronics systems based on Integrated Modular Avionics (IMA) architecture.

System designers and integrators of IMA systems are responsible for the following tasks:

- clarification and reconciliation of the software and hardware requirements with developers;
- design of the IMA platform based on the requirements for software, including:
  - distribution of software applications among available core processing modules (CPM), in compliance with required CPU cycles count, scheduling requirements for periodic applications, RAM/ROM memory usage, the bandwidth of network interfaces, etc.;
  - design of network topology based on requirements of reliability, consistency of interfaces, message delivery latency, etc;
  - verification of the IMA system model being developed for compliance with requirements defined in the project documentation of an aircraft and its individual components;
  - generation of configuration data for components of the IMA system.

To solve these problems, a system integrator of IMA systems needs a precise understanding of all the details of the system being developed , both at high and low levels of granularity, as well as utmost attentiveness when tracking the consequences of changes in the IMA system architecture. At the same time, the size of modern on-board aircraft systems and the number of essential details is so large that it is impossible to keep everything in a single person's mind.

In order to automate the process of IMA systems design and integration the MASIW Framework has been developed. The MASIW Framework is used mainly at design stages during development of IMA systems.

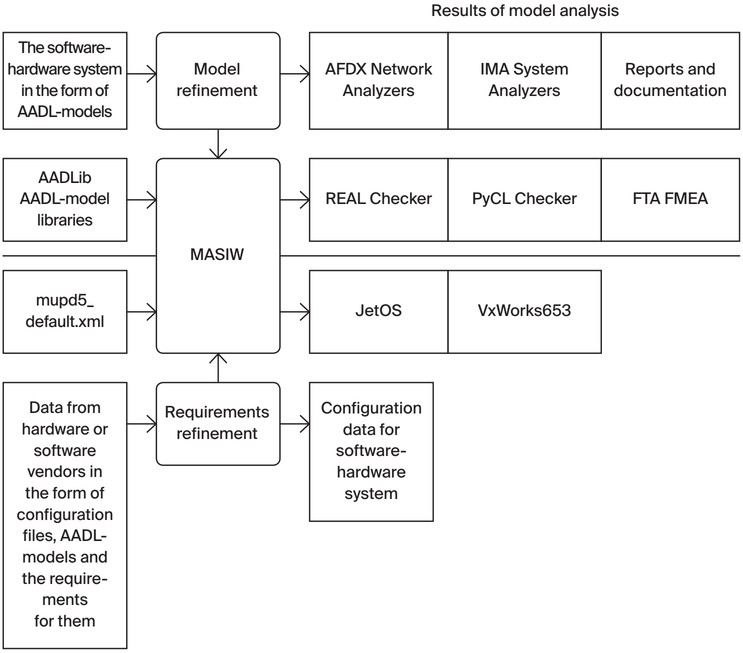
The current deployment of the MASIW Framework allows system integrator to perform the following tasks

- Creation, editing and management of models based on AADL modeling language:
  - creation and editing of models using the text and diagram editors;
  - support for team research that would enable tracking and modifying individual elements of a model;
  - support for the reuse of third-party AADL models.
- Analysis of models:
  - analysis of the hardware/software system structure: sufficiency of hardware resources, interfaces consistency, etc.;
  - analysis of data transmission characteristics of AFDX networks: message latencies, fullness of queues of the ports, etc.;
  - generation and analysis of fault trees (FTA) to determine probabilities of high-level fault events;
  - architecture-model based analysis of failures and their consequences, including generation of special descriptive tables;
  - simulation of hardware/software system model with generation of user reports including software-in-the-loop execution of on-board partitions with RTOS co-emulated with QEMU.
- Synthesis of models:
  - distribution of software applications by computational modules, taking into account limited hardware platform resources and additional restrictions on reliability and security of the hardware/software system;
- Generation of configuration data:
  - generation of schedules for processors (in particular, for ARINC-653 compatible real-time operating systems);
  - development of specialized configuration data tools, based on the provided software interface (API);
  - generation of configuration data for RTOS VxWorks653 and AFDX network equipment.

Creation, editing and management of models and configuration data are implemented using widely accepted extensions of the Eclipse environment, such as Eclipse Modeling Framework, Graphical Editing Framework, Eclipse Team Providing, SVN Team Provider, GIT Team Provider.

The MASIW Framework is modular and extendable. Third-party developers can extend the functionality of the toolset by creating their own modules to customize it.

OPERATIONAL SCHEME


Results of model analysis
The software-hardware system in the form of AADL-models
Model refinement
AFDX Network Analyzers
IMA System Analyzers
Reports and documentation
AADLib AADL-model libraries
MASIW
REAL Checker
PyCL Checker
FTA FMEA
mupd5_ default.xml
JetOS
VxWorks653
Data from hardware or software vendors in the form of configuration files, AADL-models and the require-ments for them
Requirements refinement
Configuration data for software-hardware system

CONSTRUCTIVITY 4D: THE TECHNOLOGY OF INDEXING, SEARCHING AND ANALYSIS OF LARGE SPATIO-TEMPORAL DATA

PURPOSE

The rapid growth of information volumes, as well as the need for its analysis and interpretation, leads to the development of new approaches to the management of multidimensional data and, in particular, to the management of spatio-temporal data. Usually popular general-purpose database management systems provide spatial indexing and retrieval tools for such purposes, which successfully manage processing of static information, but are not adapted for data liable to permanent changes. In turn, temporal systems are oriented to work with the data that has a history of changes, but do not take into account spatial factors. The problem of data management is even more complicated when they are not just arrays of points in a multidimensional space, but complex structures, for example, a set of mobile objects with extended boundaries and imposed composition relations. For example, managing large-scale architectural and construction programs often involves a visual analysis of millions of objects, each of which has its own geometric representation and exhibits individual dynamic behavior.

The technology developed at ISP RAS is intended to create promising software systems and services that operate large arrays of spatio-temporal data or dynamic scenes. The class of such applications is extremely wide and covers such subject areas as computer graphics and animation, geoinformatics, scientific visualization, CAD/CAM/CAE, robotics, logistics, planning and project management.

The technology provides for the usage of original methods of spatio-temporal indexing, search and analysis of data, taking into account the peculiarities of their geometric representation, complex organization and the predetermined nature of the dynamics. Support for a developed set of temporal, metric, topological and orientational operations ensures efficient execution of typical spatiotemporal queries and solution of a wide range of applied problems related to qualitative and quantitative analysis of scenes. In particular, queries for reconstructing a scene at the given point in time, retrieving objects in the given spatial region, finding nearest neighbors, determining static and dynamic collisions, and conflict-free routing in a global dynamic environment are effectively resolved.

IMPLEMENTATION

The technology is implemented as an object-oriented library in C++ language, which is an extensible set of classes, interfaces and related methods for specifying spatio-temporal data and executing typical queries to them. Due to data access virtualization, the library can be used both in the development of new software applications and in legacy applications, in order to optimize their work and expand user functions.

The organization of the library provides tools for managing data and changes, building and updating indexes, calculating and caching derived data, implementing operations, and executing applied queries. The advanced indexing system combines binary event trees, spatial decomposition trees, bounding volume trees, object cluster trees, space occupation trees. The configuration tools allow you to configure the library in the most rational way for solving applications related to special spatio-temporal queries.

The library supports the following types of operations:

- Temporal operations implement the classical interval algebra introduced by Allen with respect to time stamps of discrete events and their intervals.
- Metric operations allow you to determine the individual properties of geometric objects and the characteristics of their mutual arrangement. Diameter, area, volume, center of mass, planar projections, and distances between objects can be calculated for solid geometric objects.
- Topological operations are intended to classify the relative location of objects and establish the facts of their coincidence, intersection, coverage, touch, overlap or collision. In comparison with the known topological models DE-9IM, RCC-8, RCC-3D the operations allow constructive implementation and are applicable for the analysis of complex objects.
- Orientational operations generalize the known Frank's and Freksa's relative orientation calculi, cardinal direction calculi (CDC), oriented point relation algebra (OPRA) and are applicable for the analysis of objects with extended boundaries. This is achieved through alternative interpretations of classical directional calculi.

A computational strategy is used to determine collisions in the scenes. The strategy combines methods for the precise determination of collisions between geometric primitives, collision localization methods using spatial decomposition based on regular octrees and kd-trees, the methods of hierarchies of bounding volumes based on AABB and OBB parallelepipeds, temporal coherence methods. The collision detection strategy demonstrates uniformly high performance for scenes with different complexity characteristics.

The new original method for navigation in global dynamic environment has been developed and implemented. The method is based on extracting of spatial, metric and topological information from geometric representation of 3D scenes and its concerted usage on path planning. Global routes obtained using topological maps are verified against collisions and corrected using popular local planning algorithms like rapidly exploring random trees (RRT) and probabilistic roadmaps (PRM).

INDUSTRIAL APPLICATION

The technology has been successfully approved in the course of development of the Synchro software system intended for visual modeling, planning and management of large-scale industrial projects.

The functions of the system provide consolidation of project data and schedule, visualization of project activities, identification of collisions, the project progress monitoring, financial monitoring, preparation of illustrated documentation using a series of images and video materials. The graphic user interface of the system includes Gantt charts and synchronized views of tree-dimensional scenes, resource utilization and earned value analysis plots.

A consolidated model of project data allows to take into account and to control various factors of the project activities. For example, implemented tools for solving project planning problems in the generalized formulation (Generally Constrained Project Scheduling Problem) make it possible to generate reliable and trustfulness schedules that take into account not only the imposed time conditions, precedence relations, resource constraints and calendar rules, but also specific requirements for spatial-temporal concordance of project activities, their financial and logistic support. Examples of such requirements are conditions for attracting investment funds, restrictions on the material supply chains, rules for deployment and use of equipment, particularities of mounting elements of erected structures, conditions for reserving work areas in project sites. These requirements are important for large-scale industrial programs, in which the risks of technological and organizational errors are extremely high, and deadlines and budgets are severely limited.

Currently, the software system has been successfully applied by more than 300 companies in 36 countries.

TEXTERRA BASIC SEMANTIC ANALYZER

Smart text analysis

Texterra is a scalable platform for extracting semantics from text. It is the basic set of technologies for creating multifunctional applications. It analyzes texts using concept identification. It is included in the Unified Register of Russian software.

WHY TEXTERRA?

1 Unique combination of functions

Texterra performs a unique analysis of Russian texts based on the identification of concepts instead of just words. It differs from foreign analogues by predominant attention to Russian language. The analyzer is based on the results of basic research and provides the ability to integrate with the Elasticsearch search system, significantly expanding its capabilities. The successful combination of technologies allows the platform to compete with projects of the IBM Watson Natural Language Understanding level.

Texterra is defined by:

— High text processing speed (morphological analysis — 69 000 words per second, syntactic analysis — 39 100 words per second, coreference resolution — 10 100 words per second, full text analysis — approximately 13 600 words per second);

— Maximum attention to Russian language (unlike similar spaCy and UDPipe projects, as well as IBM Watson Natural Language Understanding, which does not support the analysis of emotions and concepts in Russian-language texts);

— Large amount of knowledge (more than 7 million concepts);

— Building the knowledge base without the involvement of experts (automatic replenishment using Wikipedia, MediaWiki, Linked Open Data, etc.);

— Scalability both in word processing speed and knowledge volume (using Apache Ignite and the original ISP RAS cloud technology);

— High text analysis accuracy due to a number of key features:
  - Multi-level search by related concepts;
  - Adaptability to slang, hashtags and errors;
  - Analysis of emotional coloring (with separation of attitude towards objects and their attributes);

  - Determination of the relationship of people and companies (based on information in the text);
  - Definition of implicit references to objects during discussions.

2 Maximum adaptability for Russian customers

Texterra — is a high-tech product that combines advanced scientific developments with the possibility of their industrial use. Our local technical support works with maximum attention to the Russian customer.

The main advantages are:

- High speed of individual solutions development;
- Two use cases:
  - as an alienable product on the customer's local server with access via both the HTTP protocol (REST architecture) and the RMI protocol;
  - o online at https://texterra.ispras.ru/;
- Continuous training of customer's developers, as well as innovative technological refinement of the product in accordance with new problems and challenges;
- Simple and fast development of specific subject areas and the ability to integrate new languages for analysis (thanks to the modern approach to machine learning).

WHO IS TEXTERRA INTENDED FOR?

- Corporate software developers (chat bots in particular);
- Developers of semantic search systems for specific subject areas (information security, medicine, auditing, etc.);
- Developers of arbitrary text processing applications.

WHO DO WE COOPERATE WITH?

Texterra was upgraded to the industrial level in the framework of cooperation with HP and Samsung (the goal of joint projects is to obtain technologies for analyzing corporate reporting and supporting the work of smart television). Currently, a number of original developments of the ISP RAS (in particular, the Talisman social media analysis technology) are working on the platform. Texterra is also used by a number of Russian government departments.

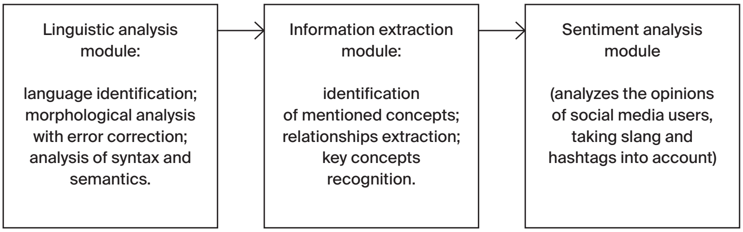
SUPPORTED LANGUAGES

Texterra analyzes texts in Russian and English.

SYSTEM REQUIREMENTS

- Any platforms supported by Java 1.8;
- At least 16 GB of RAM for each of the analyzed languages;
- We recommend using 64-bit OS.

OPERATIONAL SCHEME



Linguistic analysis module:

language identification; morphological analysis with error correction; analysis of syntax and semantics.

Information extraction module:

identification of mentioned concepts; relationships extraction; key concepts recognition.

Sentiment analysis module

(analyzes the opinions of social media users, taking slang and hashtags into account)

SOCIAL MEDIA ANALYSIS TECHNOLOGY

TALISMAN

Analyzes everything, finds the essence

Talisman — big data processing solution for social and commercial information retrieval. It recognizes patterns in relationships by analyzing large graphs from hundreds of millions of nodes.

WHY TALISMAN?

1 A unique combination of features

A unique combination of features
Talisman is an industrial solution integrated with a platform for semantic extraction (Texterra) and the original ISPRAS's technology for data mining. Considering the technological level, Talisman is comparable to the world's best analogs (Palantir Gotham and IBM Watson Content Analytics). Its advantage is the automation of routine processes utilizing the recent scientific achievements (reducing the cost of analysis).

Talisman is defined by:

- The combination of essential features, specifically:
  - Semantic analysis utilizing the capabilities of the Texterra platform (sentiment analysis, work with meaning instead of terms which are unique for the Russian language, the ability to analyze users' comments and identify implicit references to objects in discussions, etc.);
  - Analysis of large graphs consisting of hundreds of millions of nodes (including automatic construction of information distribution graphs with role definition: source, distributor, opinion leader, reader).
  - Automatic grouping of messages by topics (a map of all discussed topics in the information space, taking into account the flow between different resources);
  - Identification of true users' attributes in social networks. Determination of gender, age (to within a year), education, marital status, place of residence based on the analysis of profiles and user activity (expandable list);
  - Automatic recognition of a target audience parameters (aggregation by demographic attributes and identification of dominant values);
  - Information validation tools (detection of bots, spam filtering, and signs of audience opinion manipulation).

- Reports on monitored information within a few minutes after the publication thanks to big data analysis technologies of the Apache Hadoop stack and the elastic scalability of the system using the original cloud technology of the ISPRAS;
- Analysis of any big data: corporate, news, social networks (VK, Facebook, Twitter, Instagram, Odnoklassniki, Youtube, LinkedIn, etc.), blogs (LiveJournal), open channels of Telegram messenger and Dark web resources. Talisman can be integrated with both the original data acquisition technology of the ISPRAS and external collectors.

2 Maximum convenience for Russian customers

Talisman provides a number of profitable offers for a Russian customer:

- Industrial deployment of solutions on a customer’s equipment (Talisman is a completely detached product);
- Functioning in SAAS mode;
- Local Russian technical support (including training for customer’s developers during implementation);
- Fast adaptation and expansion of functionality for various domains (information security, medicine, auditing, etc.).

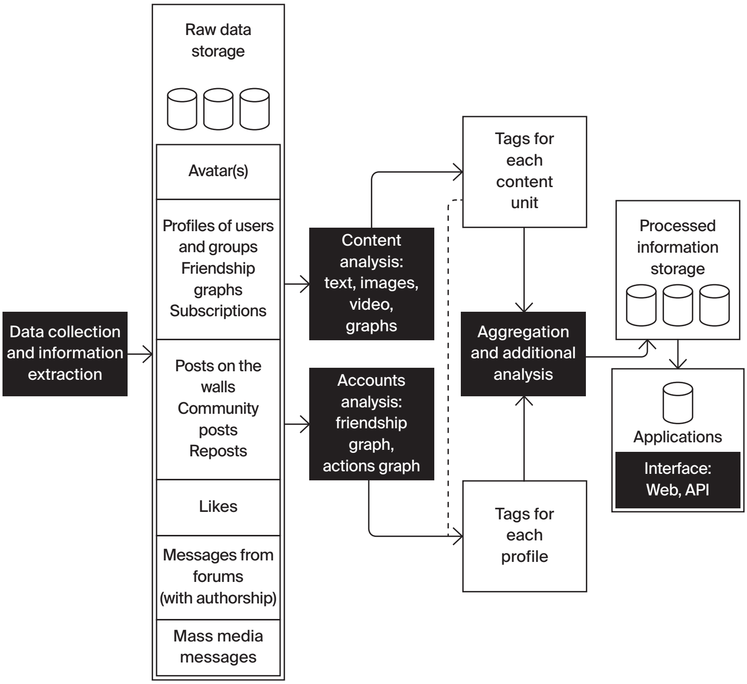
APPLICATION AREAS

- Detection of interest groups based on social media analysis e.g. target audiences (for marketing and political purposes), hotbeds of social tensions and groups addressing hot-spot issues.
- Recognition of public opinions on organizations, people and products;
- Identification of key trends and forecasting online advertisement effectiveness;
- Optimization of personnel management (efficient recruitment, data verification, assistance in developing systems of incentives based on short-term and long-term interests, leakage and disclosure of nonpublic information monitoring);
- Reputation management (in particular, determination of causes of employee and customer grievances);
- Recognition of information campaigns aiming at manipulating opinions of target audiences as well as identification of said campaign’s target audience.

SUPPORTED LANGUAGES

Talisman currently supports languages recognized by Texterra analyzer (Russian and English).

OPERATIONAL SCHEME


Raw data storage
Avatar(s)
Profiles of users and groups Friendship graphs Subscriptions
Posts on the walls Community posts Reposts
Likes
Messages from forums (with authorship)
Mass media messages
Data collection and information extraction
Content analysis: text, images, video, graphs
Accounts analysis: friendship graph, actions graph
Tags for each content unit
Tags for each profile
Aggregation and additional analysis
Processed information storage
Applications
Interface: Web, API

LINGVODOC:

GitHub for language experts

VIRTUAL LABORATORY FOR DOCUMENTATION OF ENDANGERED LANGUAGES

Lingvodoc is a system intended for collaborative multi-user documentation of endangered languages, creating multilayered dictionaries and performing scientific work with the received sound and text data. Joint project with the Institute of Linguistics of the Russian Academy of Sciences and Tomsk State University. Under development since 2012. Project website — lingvodoc.ispras.ru.

WHY LINGVODOC?

1 Unique combination of essential features

Lingvodoc is an open source, cross-platform technology (github.com/ispras/lingvodoc and github.com/ispras/lingvodoc-react), based on scientific research and combining a number of necessary functions. The system is being constantly improved. Currently, the development of a unique solution for the construction of isogloss is being finished within the framework of a joint project with the TSU.

Lingvodoc is defined by:

- Collaboration of users over vocabulary data replenishment (unlike the similar Starling project, where such work is not foreseen);
- Saving full history of users actions;
- Simultaneous work with audio and text corpuses and dictionaries based on integration with the ELAN program developed by the Max Planck Institute for Psycholinguistics (Netherlands);
- Arranging unidirectional and bidirectional links between lexical inputs within dictionaries as well as between dictionaries;
- Record, play and store annotated sounds (in wav, mp3 and flac formats), as well as construct vowel formant with subsequent visualization;
- Advanced search, which allows you to search data in dictionaries by a variety of parameters (as opposed to a similar TypeCraft project);
- The possibility of conflict-free two-way slow synchronization;
- Increased level of automation (compared with a similar Kielipankki project).

Wide opportunities for users

- Creating dictionaries of any structure, both typical two-layer (lexical inputs and paradigms layers) and multi-layer. In addition, there is an import function for ready-made dictionary structures;
- Work both with the involvement of cloud resources of the Institute for System Programming of the RAS (currently the

client-server architecture is optimized for the VMEmperor cloud infrastructure), and with the deployment of a local version with isolation of its own data;

- Availability web viewing program and desktop version;
- Open registration (with confirmation);
- Operational improvement of technology for any customer with the expansion of functionality, as well as adaptation for another scientific branch.

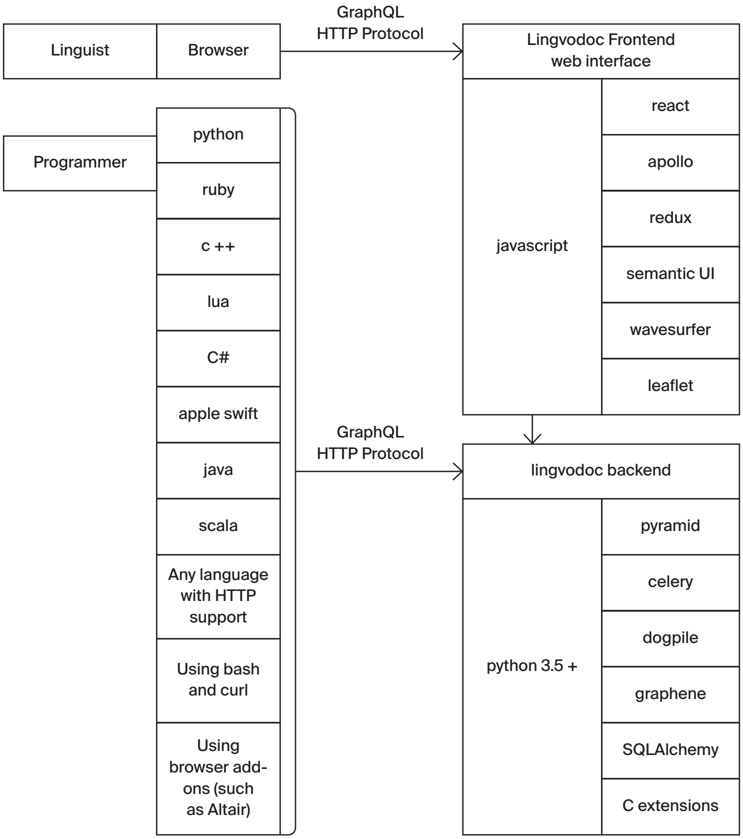
WHO IS LINGVODOC INTENDED FOR?

Lingvodoc is first of all intended for language experts who are engaging in scientific work in the field of documentation of endangered languages. It is, however, possible to adapt the technology for other purposes.

WHERE IS LINGVODOC USED?

Lingvodoc is currently used in joint projects with the Institute of Linguistics of the Russian Academy of Sciences and Tomsk State University. Negotiations are also underway with a number of research institutes.

OPERATIONAL SCHEME



SCINOON: EXPLORATORY SEARCH SYSTEM

Smart search for scientific publications

SciNoon is a system for researcher's inquiry of scientific articles. Combines a number of unique features to optimize the process of searching and analyzing the results. It particularly allows you to work in a team and keep a history of user actions. Can work with big data.

WHY SCINOON?

1 Optimal combination of essential functions

SciNoon is an innovative system designed to optimize long-term teamwork with scientific publications. It helps to explore new subject areas and maintain awareness in any one of them. It solves the same tasks as the major global analogues (Microsoft Academic Search, Google Scholar, Semantic Scholar), but at the same time it possesses unique functions.

SciNoon is defined by:

- A unique feature of supporting collaborative search required for successful teamwork:
  - common workplace accessible to all participants in the research group;
  - keeping all members of the group aware about the work of each of them by utilizing an integrated chat bot.
- Optimized system for collection of relevant scientific articles:
  - aggregation of data from different sources (user-uploaded PDF files with full texts of articles as well as metadata collected using a browser plugin integrated with Google Scholar);
  - service of recommendations based on previously selected articles;
  - citation graph available for navigation.
- Review of accumulated results via user-friendly interface:
  - visualization of all selected articles in the form of a citation graph with the scaling possibility;
  - the possibility to indicate aspects specific to the research task and the corresponding markup of articles;
  - visualization of individual articles taking their importance and indicated aspects values into account;
  - tabular presentation of selected articles taking both metadata and indicated aspects values into account;
  - support for semi-automatic clustering.

- Big data operations support:
  - own graph model to represent knowledge of all articles, authors and ongoing research. Scaling to graphs of tens of millions of nodes by using a graph database deployed on top of Apache Cassandra;
  - scaling business logic by using Akka;
  - integration with Apache Spark.

WHY SCINOON?

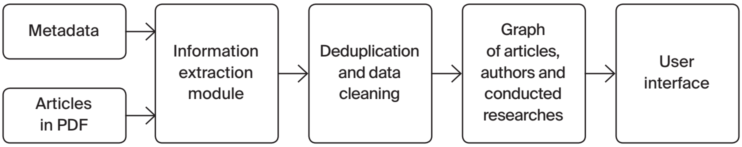
2 Maximum user convenience

- Two use cases:
  - as an alienable product on the customer’s local server;
  - online at https://scinoon.at.ispras.ru.
- Timely adaptation of technology and expansion of functionality for use in various subject areas.

WHO IS SCINOON INTENDED FOR?

- Employees of R&D departments of corporations;
- Employees of research institutes who need a tool for teamwork;
- Teachers and students of universities engaged in researcher’s inquiry for the preparation of scientific works.

OPERATIONAL SCHEME


Metadata
Articles in PDF
Information extraction module
Deduplication and data cleaning
Graph of articles, authors and conducted researches
User interface

COMPLEX OF SOLUTIONS

Maximum simplification of complex tasks

THE TECHNOLOGY OF INDEXING, SEARCHING AND ANALYSIS OF LARGE SPATIO-TEMPORAL DATA

Provides the ability to store data and perform complex, resource-intensive calculations using both containers and virtual machines. It is particularly intended for the deployment of cloud environments.

MAIN ADVANTAGES

— Adaptation flexibility to build solutions for specific problem classes (CFD, big data analytics, program analysis for defect detection);

— Technological security and alienability of solutions (the ability to recreate the infrastructure in an isolated environment with full control over it through the use of open standards, free software and scientific developments of the Institute for System Programming of the Russian Academy of Sciences);

— Operational work of local technical support, which possesses all the necessary competencies);

THE COMPLEX IS CURRENTLY REPRESENTED BY THREE SOLUTIONS:

1 Cloud environment based on the Openstack customized technology

This environment is created in the framework of a joint project with Dell company. It is designed for short-term calculations with large available resources. It is reliably operational since 2014.

— Deployed on the basis of the Openstack open technology, which is the standard for building large cloud systems;

— Provides users with all the functionality necessary for short-term calculations with large available resources:
  - management of virtual networks and computer clusters using Keystone, Neutron and Nova systems (similar to Amazon EC2);
  - block data storage based on the Cinder system (similar to Amazon Elastic Block Storage);
  - easily expandable object storage based on Openstack Swift (similar to Amazon S3).

— Provides the possibility to develop and implement various services at the PaaS level:
  - Big Data Open Lab computer cluster for big data analyzing with fully configured Apache Spark, Apache

  - Hadoop and Apache Ignite systems and an arbitrary number of computing nodes (starting one cluster takes about 5 minutes). It is publicly available (https://github.com/ispras/spark-openstack);
  - for artificial intelligence research using Tensorflow, Caffe, etc., as well as modern hardware (NVIDIA Tesla V100 servers on the SXM2 bus);
  - to work with HPC.

2 VMEmperor virtual machine management solution

Designed in the Institute for System Programming of the Russian Academy of Sciences for solving internal problems, publicly available (https://github.com/ispras/vmemperor). Designed to manage virtual resources at the IaaS level. It has been continuously running on the XCP-ng / Citrix XenServer platform since 2012 providing users with easy access to virtual resources and their orchestration.

3 Fanlight web-laboratories organization platform

Created as a result of the Institute for System Programming of the Russian Academy of Sciences participation in the «University Cluster» program and in the Open Cirrus international project (established by Hewlett-Packard, Intel and Yahoo!). It is intended for deploying SaaS infrastructures for web-based computing labs using Docker Compose. It is built on virtual containers and operates on the basis of virtual desktops in the DaaS model (Desktop as a Service). The platform is available for users on the fanlight.ispras.ru website and supports applications developed for Linux kernel based OS only.

- Demonstrates high performance cloud computing through the use of containers:
  - comfortable work with heavy CAD-CAE engineering applications that require hardware acceleration support for 3D graphics for complex visualization;
  - support for running MPI, OpenMP, CUDA applications by accessing HPC clusters, multi-core processors and NVIDIA graphics accelerators.
- Expands computing capabilities at the PaaS level by engaging hardware resources (HPC / BigData clusters, storage systems, servers with graphic accelerators);
- Allows you to perform customization for a given application area by integrating specialized design application packages. There is a particular experience of implementation in:
  - the MCC field: OpenFOAM, SALOME, Paraview, etc.;
  - the Gas&Oil field: tNavigator, Eclipse, Roxar, Tempest, etc.
- Allows the user to work using any thin client (including mobile devices) without auxiliary software;
- It can be deployed on a server, a computing farm, in the cloud (from IaaS level) or in its own data processing center.

IMPLEMENTATION EXPERIENCE

The Big Data Open Lab computer cluster is used to analyze information flows in the Talisman social media analysis technology and to support operation of other technologies of the Ivannikov Institute for System Programming of the Russian Academy of Sciences (in particular, to analyze Android OS using Svace). A joint project with Huawei (large graphs analysis using big data processing technologies) and the Tizen OS lifecycle support infrastructure that allows organizing the OS components joint development process and automate the regular assembly and testing of samples are implemented. In addition, a number of works are carried out with the participation of the Ministry of Science of the Russian Federation.

The capabilities of the Fanlight platform were used in a number of joint projects for the deployment of web-laboratories with Russian Federal Nuclear Center of the All-Russian Scientific Research Institute of Experimental Physics, OOO RRS-Baltika, Keldysh Institute of Applied Mathematics (development of technology to increase and efficiently use the hydrocarbon raw materials resource potential of the Union State) as well as the Laboratory of Continuum Mechanics (https://unicfd.ru).

VMEmperor was not used in external commercial projects, however, it is used in internal projects of the Ivannikov Institute for System Programming of the Russian Academy of Sciences.