RESEARCH CENTRE

Inria Centre at the University of Lille

IN PARTNERSHIP WITH:

Université de Lille, Berger-Levrault

2023

ACTIVITY REPORT

Project-Team

EVREF

Reflective Evolution of Ever-running Software Systems

IN COLLABORATION WITH: Centre de Recherche en Informatique, Signal et Automatique de Lille

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed programming and Software engineering

Inria

Contents

Project-Team EVREF

Creation of the Project-Team: 2023 April 01

Keywords

Computer sciences and digital sciences

A2. – Software

A2.1. – Programming Languages

A2.1.3. – Object-oriented programming

A2.1.10. – Domain-specific languages

A2.1.12. – Dynamic languages

A2.5. – Software engineering

A2.5.3. – Empirical Software Engineering

A2.5.4. – Software Maintenance & Evolution

A2.5.5. – Software testing

A2.6. – Infrastructure software

A2.6.3. – Virtual machines

A2.6.4. – Ressource management

Other research topics and application domains

B6. – IT and telecom

B6.1. – Software industry

B6.1.1. – Software engineering

B6.1.2. – Software evolution, maintenance

B6.5. – Information systems

1 Team members, visitors, external collaborators

Research Scientists

- Stéphane Ducasse [Team leader, INRIA, Senior Researcher, HDR]
- Nicolas Anquetil [UNIV LILLE, Associate Professor Detachement, HDR]
- Christophe Bortolaso [Berger-Levrault, Senior Researcher]
- Steven Costiou [INRIA, Researcher]
- Marcus Denker [INRIA, Researcher]
- Nicolas Hlad [Berger-Levrault, Researcher]
- Guillermo Polito [INRIA, Researcher]
- Larisa Safina [INRIA, ISFP, from Nov 2023]
- Anas Shatnawi [Berger-Levrault, Researcher]
- Benoit Verhaeghe [Berger-Levrault, Researcher]

Faculty Members

- Anne Etien [UNIV LILLE, Professor, HDR]
- Imen Sayar [UNIV LILLE, Associate Professor, from Sep 2023]

PhD Students

- Nour Jihene Agouf [AROLLA, CIFRE]
- Valentin Bourcier [INRIA]
- Santiago Bragagnolo [Berger-Levrault, until May 2023, CIFRE]
- Gabriel Darbord [INRIA]
- Aless Hosry [INRIA]
- Sebastian Jordan Montaño [INRIA, from Oct 2023]
- Soufyane Labsari [INRIA, from Dec 2023]
- Nahuel Palumbo [INRIA]
- Iona Thomas [INRIA]
- Maximilian Willembrinck Santander [INRIA, until Nov 2023]

Technical Staff

- Christophe Demarey [INRIA, Engineer, 60%]
- Cyril Ferlicot-Delbecque [INRIA, Engineer]
- Soufyane Labsari [INRIA, Engineer, until Nov 2023]
- Esteban Lorenzano [INRIA, Engineer, Pharo Consortium]
- Milton Mamani Torres [INRIA, Engineer]

- Hernan Federico Morales [INRIA, Engineer]
- Larisa Safina [INRIA, Engineer, until Oct 2023]
- Pablo Tesone [INRIA, Engineer, Pharo Consortium]
- Clotilde Toullec [INRIA, Engineer]

Interns and Apprentices

- Enzo Demeulenaere [INRIA, Apprentice, from Oct 2023]
- Pol Durieux [INRIA, Intern, from Nov 2023]
- Sebastian Jordan Montano [INRIA, Apprentice, from Apr 2023 until Aug 2023]
- Patricia Totoum Mandoum [INRIA, Intern, from May 2023 until Aug 2023]
- Adrien Vanegue [INRIA, Apprentice, from Apr 2023]
- Thomas Wattebled [UNIV LILLE, Intern, from Apr 2023 until Jul 2023]

Administrative Assistant

- Aurore Dalle [INRIA]

Visiting Scientists

- Mihajasoa Léa Fanomezana [Université de Fianarantsoa, Madagascar, from Sep 2023]
- Balsa Sarenac [University of Novi Sad, Serbia, from Aug 2023 until Sep 2023]

2 Overall objectives

The objectives of EVREF are to study and support the continuous evolution of large software systems in a holistic manner following three main axes: (1) analyses and approaches for migration and evolution of existing (legacy) systems, (2) new tools to support daily evolution, and (3) infrastructure to build language runtimes to support software evolution, new tools, frugal systems and security language features. In the context of the first axis, we proposes a specific research agenda with Berger-Levrault R&D.

Evolving large software systems is a challenge. Decades of academic research have *somehow* produced tools and platforms that help companies to maintain their systems. But keeping legacy systems active and relevant is still a really complex task. An aggravating challenge is that some of these systems can never stop (production lines, wafer production systems, auction managers, etc) and need to be updated while running at production sites. In addition, because the production environment is not the same as the development environment, the only way to spot and fix a bug is often by directly accessing software *in production, while running.*

Supporting the evolution of such *ever-running* systems is an important challenge for our industry because it must deal with more and more changing requirements and the need for dynamic adaptation. To address this challenge EVREF works on (1) *analyses and approaches based on language-specific metamodels and their accompanying processes* such as test generation, semi-automated migration, or business rule identification; (2) *new generation debuggers, profilers and tools for reverse engineering* — we will tackle new areas such as the support for non-functional requirements (robustness, memory consumption, ...) —, (3) *language and runtime infrastructure to support evolution, green computing, security, and tooling* as a step towards self-evolvable runtimes. The EVREF approach is reflective in the sense that by controlling the underlying execution engine it will explore different facets of evolution and tooling.

3 Research program

EVREF is built around a holistic vision of the eternal software challenge. It acknowledges that we need to be able to work on different levels to support the evolution of software under different scenarios. The fact that we will work on a full stack (still making progress in each area) creates a situation where the team will be in the position to think and propose solutions that would not be possible otherwise. The reflective stress in the project title is that the axes can reflect and influence each other and can help each other in client/provider of problems or solutions.

The agenda defined with Berger-Levrault acts a reality ground for this research agenda. The evolution challenges faced by Berger-Levrault are still unresolved challenges that any software company has to struggle with: testing to control migration, migration to new technology, business rule identification and software maps are key challenges. They do not imply that the software is running and that migration should happen while the system is executing but they are typical scenarios.

3.1 Research Axes within EVREF

The research axes in EVREF are built to form an articulated whole around the challenges of evolution of constantly changing and running systems. The three axes are interconnected often in client relationships, *e.g.,* profilers requires low-level information provided by virtual machines but virtual machines requires advanced profilers. Controlling virtual machines opens the doors of many possibilities both at the level of tools but also language design for isolation or security.

- **Axis 1 – Evolution of Ever-running Systems.** This axis is about how to effectively evolve large and complex software. This covers a large spectrum of topics such as visualisation, metrics, analysis,... This includes for example migration from one language to other one or from a library version to another one. This is within this axis that the team will work in partnership with Berger-Levrault. The axis is built around the Moose platform and its current redesign effort.

- **Axis 2 – New Generation Tools for Daily Tasks.** This axis is about how to offer advanced tools for everyday development: it focuses on debuggers, profilers and tools to reverse engineer code. It follows the work around debugging started in RMOD.

- **Axis 3 – A Generative Approach to Modular and Versatile Virtual Machines.** This axis is about how to improve the building of virtual machines to support their exploration and application to tools, security, green computing, ... This axis is also providing infrastructure for the other axes. The exploratory action is an important basis for this axis. In addition interactions with the Pharo consortium engineers and the use of the industry level Pharo virtual machine will naturally take place.

There are possible and welcomed overlaps between areas covered in the interaction with Berger-Levrault: for example transpilation is the basis of the Pharo VM compilation tool chain, migration is a topic of interest for Berger-Levrault. Still we list such item in the axis because the research agenda of EVREF is larger than its interaction with Berger-Levrault. A cross-fertilisation one the same topic will happen but without one taking over the others.

The third axis, *A Generative Approach to Modular and Versatile Virtual Machines*, will also support the other axes by exposing specific runtime information (such as exposing Polymorphic inline caches, possibility of instrument object creation,...) or offering the possibility to extend the virtual with new or modified low-level functionality. It will also take into account the needs and feedback from the tool builders.

3.2 Axis 1 – Evolution of Ever-running Systems

Supporting software evolution is an important and challenging topic inherently linked to software. Indeed software models the world and the world is changing. Therefore software evolution is ineluctable. In EVREF we will work on fundamental aspects of software evolution:

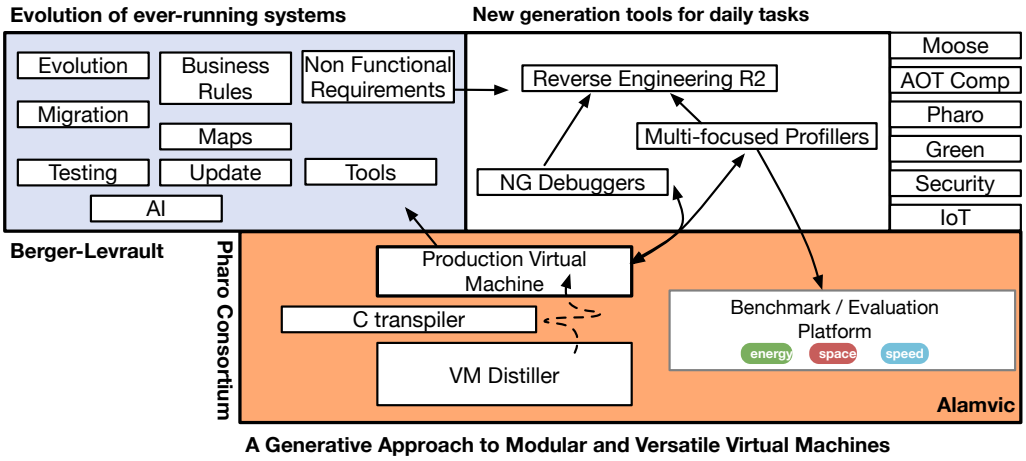

Figure 1: EVREF vision: Three interacting axes.

- **Towards automatic evolution.** We will work on supporting semi-automated evolution in the case of libraries update. We will extend our work and support both the library developers and their users to migrate to more recent versions by analysing past activities and learning automated rules.

- **Migration.** We will enhance our metamodel-based approach of front-end migration to support interlanguage migration.

- **Non-functional requirement identification and extraction.** To help developers during their maintenance task we will take into account non-functional requirements (NFR) and propose software maps and reverse engineering techniques to reveal such hidden software aspects.

- **NFR aware code transformations.** We will extend refactorings to support domain specific and non-functional requirements.

3.3 Axis 2 – New Generation Tools for Daily Tasks

We will work on tools to support developers with a focus on improving daily development tasks.

- **New Generation Debuggers** will propose new debugging techniques such as object centric and back in time debuggers.

- **Multi-Layered profilers** will rethink profilers and how systems are benchmarked.

- **Reverse engineering revisited** will revisit reverse engineering techniques taking into account non-functional requirements such as memory consumption, security concerns and others.

3.4 Axis 3 – A Generative Approach to Modular and Versatile Virtual Machines

Virtual machines are a key assets both from an engineering and research point of view. As extremely complex pieces of software (garbage collector, multi layered interpreters, speculative inliner), they raise the question of their definition, construction and validation. As a research vehicule they are keys for innovation at the level of language design, security, ever-running systems, or green computing. This axis is based on the work the team did together with the Pharo consortium and the support of the Alamvic Inria Exploratory Action led by G. Polito.

Main Objective. EVREF will explore how Virtual Machines are designed as a **whole**, and how they are optimised for a *large range of concerns* that include not only execution speed but also energy and space consumption, for applicability in *security, green computing, IoT and robotics.* Such research effort will take place in the context of the Pharo virtual machines and its associated production chain.

- **A transpilation chain.** Based on our current architecture, we will design a transpilation chain that will take into account heuristics (memory, concurrency, shipset, speed).
- **Metamodeling and DSL for VM building.** VM optimisations are complex and spread over many aspects of the logic, we will evaluate how such optimisations can be represented and extracted to be recomposed using a domain specific language.
- **JIT compilers and optimizers.** Building modern Just in time compilers and native dynamic optimizer is a difficult task but key to support modern language execution, we want to assess the design and architecture of alternative dynamic optimizer.
- **New Evaluation Methodologies.** A VM is a complex piece of software with adaptative behavior we will work on ways to measure performance to be able to gather actionable information.

4 Application domains

4.1 Programming Languages and Tools

Many of the results of EVREF are improving programming languages or development tools for such languages. As such the application domain of these results is as varied as the use of programming languages in general. Pharo, the language that EVREF develops, is used for a very broad range of applications. From pure research experiments to real world industrial use (the Pharo Consortium has more than 25 company members).

Examples are web applications, server backends for mobile applications or even graphical tools and embedded applications

4.2 Software Reengineering

Moose is a language-independent environment for reverse and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization. As such Moose is used for analyzing software systems to support understanding and continuous development as well as software quality analysis.

5 Social and environmental responsibility

5.1 Footprint of research activities

The main environmental footprint of EVREF is related to international travel. Meeting researchers in person is indispensable for research.

We try to reduce travel by using online meetings as much as possible. The team tries to reduce impact of daily local travel by the use of local transport and biking to work.

5.2 Impact of research results

Our work on language runtimes has the potential to reduce energy consumption.

Reengineering can be understood as a kind of *"recycling"*. Our tools allow companies to use systems for a longer time, reducing environmental impact of software that is created as a new project.

All software we develop as part of our research is released as Open Source, all our publications are available in the HAL archive.

6 Highlights of the year

- Creation of the Project-Team EVREF with Berger-Levrault
- Imen Sayar joined the team as Associate Professor, September 2023
- Larisa Safina joined the team on an Inria Starting Faculty Position (ISFP), November 2023
- Release of Pharo 11.

6.1 Awards

- Prix FIEEC-Bpifrance de la Recherche appliquée 2023 (Stéphane Ducasse)
- ESUG Innovation Technology Awards 19th Edition: 3d price for "The Pharo Debugger and Debugging Tools". Steven Costiou, Adrien Vanègue, Valentin Bourcier, and the Pharo team.
- Best paper award: *Garbage Collector Tuning in Pathological Allocation Pattern Applications.* IWST'23 (Nahuel Palumbo, Sebastian Jordan Montaño, Guillermo Polito, Pablo Tesone and Stéphane Ducasse) [12]
- VISSOFT 2023 Most-influential paper Award for: *Performance Evolution Blueprint: Understanding the Impact of Software Evolution on Performance.* (Stéphane Ducasse, Marcus Denker) [27]

7 New software, platforms, open data

7.1 New software

7.1.1 Moose

Name: Moose: Software Analysis and Re-engineering Platform

Keywords: Software engineering, Meta model, Software visualisation, Parsing, Software quality, Code analysis

Scientific Description: Moose is a program manipulation platform based on a generic meta-model of programming languages.

A collection of atomic properties of programming languages (eg. an entity has a name, it can be invoked, it has a type, ...) allows to build specialized meta-models for each programming language.

The Moose analysis platform is based on these atomic properties to offer generic tools independent of the programming languages handled.

Functional Description: Moose is an open and extensible platform for software analysis and re-engineering.

It integrates language models, metrics, analysis algorithms, and visualization and navigation engines. Moose's development has been estimated at 200 man-years.

URL: `https://modularmoose.org`

Publication: hal-02972159v1

Contact: Stephane Ducasse

Participants: Anne Etien, Nicolas Anquetil, Stephane Ducasse

Partners: Université de Berne, Sensus, Pleiad, USI, Vrije Universiteit Brussel, Berger-Levrault

7.1.2 Pharo

Name: The platform Pharo

Keywords: Live programmation objet, Reflective system, Web Application, Test, Virtual Machine Image, Object-Oriented Programming

Functional Description: Pharo is a pure object reflective and dynamic language inspired by Smalltalk. In addition, Pharo comes with a full advanced programming environment developed under the MIT License. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo's goal is to be a platform to build and deploy mission critical applications, while at the same time continue to evolve.

Release Contributions: Better, faster, cleaner

URL: http://www.pharo.org

Contact: Marcus Denker

Participants: Christophe Demarey, Damien Pollet, Esteban Lorenzano, Marcus Denker, Stephane Ducasse, Guillermo Polito, Pablo Tesone

Partners: BetaNine, Reveal, Inceptive, Netstyle, Feenk, ObjectProfile, GemTalk Systems, Greyc Université de Caen - Basse-Normandie, Université de Berne, Yesplan, RMod, Sensus, Université de Bretagne Occidentale, École des Mines de Douai, ENSTA, Uqbar foundation Argentina, ZWEIDENKER, LifeWare, JPMorgan Chase, KnowRoaming, ENIT, Spesenfuchs, FINWorks, Esug, FAST, Ingenieubüro Schmidt, Projector Software, HRWorks, Inspired.org, Palantir Solutions, High Octane, Soops, Osoco, Ta Mère SCRL, University of Yaounde 1, Software Quality Laboratory, Software Institute Università della Svizzera italiana, Universdad Nacional de Quilmes, UMMISCO IRD, Université technique de Prague

8 New results

We present the results of the year for the three axis of EVREF.

8.1 Evolution of Ever-running Systems

Pharo: a reflective language - A first systematic analysis of reflective APIs.

Participants: Iona Thomas, Stéphane Ducasse, Pablo Tesone, Guillermo Polito.

Reflective operations are powerful APIs that let developers build advanced tools or architecture. Reflective operations are used for implementing tools and development environments (e.g., compiler, debugger, inspector) or language features (e.g., distributed systems, exception, proxy, aspect-oriented programming). In addition, languages are evolving, introducing better concepts, and revising practices and APIs. As such, since 2008 Pharo evolved and was built on Squeak which changed from the original Smalltalk reflective APIs. Pharo has one of the largest reflective feature sets ranging from structural reflection to on-demand stack reification. In addition, new metaobjects got integrated such as first-class instance variables. Finally, reflective facilities are mixed with the base-level API of objects and classes and reflective features are heavily used by the system tools. Getting an understanding of such API is, however, tedious when the API is large and evolved over a decade. There is a need for a deep analysis of current reflective APIs to understand their underlying use, potential dependencies, and whether some reflective features can be scoped and optional. In this article, we analyze the reflective operations used in Pharo 11. We classify the current reflective operations in different families. Also, we identify a set of issues raised by the use of reflective operations. Such an analysis of reflective operations in Pharo is important to support

the revision of the reflective layer and its potential redesign. [15], [25]

External dependencies in software development.

Participants: Aless Hosry, Nicolas Anquetil.

Successful software requires constant modifications. To guarantee the continuous proper functioning of the applications, developers need to understand them well, particularly by having an accurate map of the dependencies between the parts they are modifying. However, some of these dependencies are not easily identified. For example, in an Android application, there are dependencies between the Java source code and XML parts, some of which are materialized by a generated "R" Java class. We call such dependencies external because they are introduced by some agent external to the source code. On top of the categorization of dependencies defined in the literature, we define restrictions on the External Dependencies that allow us to verify the source code and identify possible flaws. We created a common approach relying on reusable patterns to search for containers and entities that are part of such dependencies and implemented it in a prototype that we validate on two different projects from GitHub and developed using different frameworks. [8]

Interactive, Iterative, Tooled, Rule-Based Migration of Microsoft Access to Web Technologies.

Participants: Santiago Bragagnolo, Nicolas Anquetil, Stéphane Ducasse.

In the context of a collaboration with Berger-Levrault, an IT company producing information systems, we are working on migrating Microsoft Access monolithic applications to the web front-end and microservices back-end. Like in most software migrations, developers must learn the target technology, and they will be in charge of the evolution of the migrated system in the future. To respond to this problem, we propose the developers take over the migration project. To enable the developers to drive the migration to the target systems, we propose an Interactive, Iterative, Tooled, Rule-Based Migration approach. The contributions of this article are (i) an iterative, interactive process to language, library, GUI and architectural migration; (ii) proposal of a set of artefacts required to support such an approach; (iii) three different validations of the approach: (a) library and paradigm usage migration to Java and Pharo, (b) tables and queries migration to Java and Typescript, (c) form migration to Java Springboot and Typescript Angular. [21]

Telemetry of Legacy Web Applications: An Industrial Case Study.

Participants: Anas Shatnawi, Gabriel Darbord, Nicolas Bortolaso.

Berger-Levrault, like many companies, has legacy web applications that still bring great values, and cannot be easily replaced. To maintain these applications, it needs data about user navigation, backend actions and client-server data exchange. Berger-Levrault has relied on a traditional logging approach that partially collects these data, requires modifying the application code and heavily impacts its performance. To address the limitations of this logging approach, we propose to replace it by a modern software telemetry approach. Existing telemetry approaches do not meet our needs, they should be extended based on our objectives, technological constraints and industrial regulations. We report our experience in instrumenting real, large-scale, industrial legacy web applications based on a telemetry approach. Our goal is to automatically instrument legacy web applications to collect data fulfilling our industrial needs. We extend the automatic instrumentation capabilities of OpenTelemetry agents to instrument our applications without modifying their code. We define a telemetry architecture to integrate

telemetry components with legacy web applications. Also, we empirically evaluate the performance overhead produced by our agents. The results show that there is no significant overhead when using OpenTelemetry agents. However, this overhead is sensitive to the size of data being serialized when instrumenting client-server data exchange. Moreover, we discuss lessons learned about the technical challenges we faced during the industrialization of our approach. [23]

Parsing Fortran-77 with proprietary extensions.

Participants: Larisa Safina, Stéphane Ducasse, Nicolas Anquetil.

Far from the latest innovations in software development, many organizations still rely on old code written in "obsolete" programming languages. Because this source code is old and proven, it often contributes significantly to the continuing success of these organizations. Yet to keep the applications relevant and running in an evolving environment, they sometimes need to be updated or migrated to new languages or new platforms. One difficulty of working with these "veteran languages" is being able to parse the source code to build a representation of it. Parsing can also allow modern software development tools and IDEs to offer better support to these veteran languages. We initiated a project between our group and the Framatome company to help migrate old Fortran-77 with proprietary extensions (called Esope) into more modern Fortran. We explain how we parsed the Esope language with a combination of island grammar and regular parser to build an abstract syntax tree of the code. [24]

Implementation-First Approach of Developing Formal Semantics of a Simulation Language in VDM-SL.

Participants: Stéphane Ducasse.

Formal specification is a basis for rigorous software implementation. VDM-SL is a formal specification language with an extensive executable subset. Successful cases of VDM-family including VDM-SL have shown that producing a well-tested executable specification can reduce the cost of the implementation phase. This paper introduces and discusses the reversed order of specification and implementation. The development of a multi-agent simulation language called RE:MOBIDYC is described and examined as a case study of defining a formal specification after initial implementation and reflecting the specification into the implementation code. [11]

A Visualization for Client-Server Architecture Assessment.

Participants: Nour Jihene Agouf, Soufyane Labsari, Stéphane Ducasse, Anne Etien, Nicolas Anquetil.

Maintaining large legacy systems often requires understanding their architecture. This is important since legacy system architecture decay over time and architecture violations may dramatically impact planned renovation actions. Merely reading source files is time-consuming and often highly inefficient. Visualizations have been proposed as a tool to support architecture understanding. Some software architecture visualizations decompose the software system architecture into layers, components, or slices from a structural viewpoint. Such visualizations, however, do not take into account the specificities of clientserver applications. They do not help maintainers identify and understand software architecture violations. We propose CLISERVO, a new visualization to help software maintainers detect architectural violations in client-server systems. CLISERVO classifies client-server entities into different levels of

dependencies, shared entities, or ambiguous entities (e.g., entities that belong abnormally to different layers). CLISERVO identifies and presents entities in their corresponding layers from two distinct viewpoints: global overview entities and violations, i.e ambiguous entities and illegal dependencies between layers. We validated our approach on three real-world industrial projects with access to their maintainers. We report the findings of 91 ambiguous entities, 29 purportedly shared and idle entities, 24 and 82 elements defined as shared but only used by the client or server, and 12 relations violating the layered architecture.[4]

JavaBIP meets VerCors: Towards the Safety of Concurrent Software Systems in Java.

Participants: Larisa Safina.

We present "Verified JavaBIP", a tool set for the verification of JavaBIP models. A JavaBIP model is a Java program where classes are considered as components, their behaviour described by finite state machine and synchronization annotations. While JavaBIP guarantees execution progresses according to the indicated state machine, it does not guarantee properties of the data exchanged between components. It also does not provide verification support to check whether the behaviour of the resulting concurrent program is as (safe as) expected. This paper addresses this by extending the JavaBIP engine with run-time verification support, and by extending the program verifier VerCors to verify JavaBIP models deductively. These two techniques complement each other: feedback from run-time verification allows quicker prototyping of contracts, and deductive verification can reduce the overhead of run-time verification. We demonstrate our approach on the "Solidity Casino" case study, known from the VerifyThis Collaborative Long Term Challenge. [5]

A VM-Agnostic and Backwards Compatible Protected Modifier for Dynamically-Typed Languages.

Participants: Iona Thomas, Stéphane Ducasse, Guillermo Polito, Pablo Tesone.

In object-oriented languages, method visibility modifiers hold a key role in separating internal methods from the public API. Protected visibility modifiers offer a way to hide methods from external objects while authorizing internal use and overriding in subclasses. While present in main statically-typed languages, visibility modifiers are not as common or mature in dynamically-typed languages. In this article, we present ProtDyn, a self-send-based visibility model calculated at compile time for dynamically-typed languages relying on name-mangling and syntactic differentiation of self vs non self sends. We present #Pharo, a ProtDyn implementation of this model that is backwards compatible with existing programs, and its port to Python. Using these implementations we study the performance impact of ProtDyn on the method lookup, in the presence of global lookup caches and polymorphic inline caches. We show that our name mangling and double method registration technique has a very low impact on performance and keeps the benefits from the global lookup cache and polymorphic inline cache. We also show that the memory overhead on a real use case is between 2% and 13% in the worst-case scenario. Protected modifier semantics enforces encapsulation such as private but allow developers to still extend the class in subclasses. ProtDyn offers a VM-agnostic and backwards-compatible design to introduce protected semantics in dynamically-typed languages. [3]

Migrating the Communication Protocol of Client-Server Applications.

Participants: Gabriel Darbord, Benoît Verhaeghe, Anne Etien, Nicolas Anquetil.

As part of a collaboration with Berger-Levrault, an international IT company, we are working on the migration of client-server applications. To escape legacy technologies and to evolve towards a "software as a service" model, the company decided to migrate the client side of its applications to Angular 14 and the server side to Spring Boot. In this paper, we focus on the migration of client-server communication from RMI and GWT-RPC to the REST architectural style. We identify issues associated with such a migration and propose a tool-based approach to address them. The migration involves (1) identifying existing services and exchanged data structures; (2) migrating the services; (3) migrating the data structures on the new client side; and (4) in some cases, reducing the amount of exchanged data to address performance issues. We experimented with our approach on four of the company's applications currently using RMI or GWT-RPC. [2]

8.2 New Generation Tools for Daily Tasks

ILLIMANI Memory Profiler.

Participants: Sebastian Jordan Montaño, Guillermo Polito, Stéphane Ducasse, Pablo Tesone.

Modern programming languages provide automatic memory management with an efficient garbage collector making the memory management of an application transparent to the developer. There is a need for practical tools to support developers in their understanding of the memory consumption of their applications. In this paper, we present a prototype version of ILLIMANI: a precise object allocation profiler. It has a rich object model that provides information about the objects' allocation context, the evolution of memory usage, and garbage collector stress. We were able to find an object allocation site in the class UITHEME that was making 99,9% redundant allocations. We developed a Color Palette cache at the domain level that removed all the redundant allocations. [22]

Improving Performance Through Object Lifetime Profiling: the DataFrame Case.

Participants: Sebastian Jordan Montaño, Nahuel Palumbo, Guillermo Polito, Stéphane Ducasse, Pablo Tesone.

Being capable of profiling the object lifetimes of an application gives information that can be used to optimize the GC performance and improve overall execution time. One can pre-tenure objects based on profiler information, tune the GC parameters, or take decisions about pre-allocating bigger memory segments. However, accessing object lifetimes is difficult because it requires monitoring any object GC reclamation. We developed an open-source lifetime profiler. Our current implementation does not require Virtual Machine modification. It is based on ephemerons and method proxies. We profiled DataFrame and we observed a significant number of objects that lived a long time. We used this information to tune the garbage collector parameters and we got up to 6.8 times of performance improvements. [10]

Debugging Video Games: A Systematic Mapping.

Participants: Valentin Bourcier, Steven Costiou.

The video game industry is a vast and lucrative sector that generates significant revenue. However, a recurring issue within this industry is the release of games with numerous bugs. The process of debugging, a challenging and expensive undertaking in software development, becomes crucial in rectifying these issues. It is worth noting that such bugs can significantly impact the commercial performance of games in the market. Therefore, an imperative arises to foster further academic research that presents dedicated

debugging techniques aimed at enhancing the process specifically within the realm of video game development. In this paper, we employ a systematic mapping study methodology to discern the existing body of knowledge concerning debugging practices for video games. Through a systematic selection process, we selected 21 relevant studies for analysis, enabling the synthesis of data and facilitating the creation of a overview that encapsulates the state of the art on debugging video games. We identified work analyzing challenges of debugging games, proposing bug detection techniques or dedicated debugging tools. However, these work are always bound to specific contexts or kinds of games. Our analysis shows that there is no academic body of knowledge about debugging video games. Thus, this initial exploration not only raises pertinent questions but also presents prospects for conducting empirical and controlled experiments to enhance our understanding of effective debugging strategies within the context of video game development. We complete our analysis with a discussion of testing and debugging techniques, and how they could help and open new research opportunities for the debugging of video games. [16]

A Unit Test Metamodel for Test Generation.

Participants: Gabriel Darbord, Anne Etien, Nicolas Anquetil, Benoît Verhaeghe.

Unit testing is a crucial aspect of software development, but developers often lack the time and resources to create comprehensive tests for their code. This can result in codebases that are vulnerable to bugs and issues. To address this problem, we present a unit test metamodel that enables the generation of unit tests. The metamodel provides a language-agnostic abstraction that enables automated transformation and code generation. We use the Famix family of models and tools from the Moose platform to build our metamodel and for code analysis. To generate realistic tests, we plan to use application traces consisting of method arguments and results. Our objective is to generate maintainable, human-readable tests that cover important use cases, including edge cases and rare scenarios. In this paper, we discuss related work in unit test generation, present the details of our metamodel, including its design, implementation and usage, and explore future work that will evaluate the effectiveness of our approach through case studies and experiments. [6]

A manual categorization of new quality issues on automatically-generated tests.

Participants: Nicolas Anquetil.

Diverse studies have analyzed the quality of automatically generated test cases by using test smells as the main quality attribute. But recent work reported that generated tests may suffer a number of quality issues not necessarily considered in previous studies. Little is known about these issues and their frequency within generated tests. In this paper, we report on a manual analysis of an external dataset consisting of 2,340 automatically generated tests. This analysis aimed at detecting new quality issues, not covered by past recognized test smells. We use thematic analysis to group and categorize the new quality issues found. As a result, we propose a taxonomy of 13 new quality issues grouped in four categories. We also report on the frequency of these new quality issues within the dataset and present eight recommendations that test generators may consider to improve the quality and usefulness of the automatically generated tests. [7]

Pattern matching in Pharo.

Participants: Aless Hosry, Nicolas Anquetil, Stéphane Ducasse.

Pattern matching is a computational technique used to identify and analyse recurring structures or patterns within data and enable the extraction of meaningful information from the data. In the field of software engineering, pattern matching is often used in applications such as compilers and linters, where the ability to recognise and understand structural patterns in source code is essential for tasks such as optimisation, error detection, and code refactoring, but pattern matching also finds its use for querying deeply recursive structures. Pharo already offers RBParseTreeSearcher, a powerful Domain Specific Language (DSL) defining its own textual syntax with a focus on matching the Pharo Abstract Syntax Tree (AST). This tool is the base of the Pharo Refactoring Framework. However, RBParseTreeSearcher is dedicated to Pharo AST matching only and cannot perform on other kinds of structures or ASTs. In this paper, we introduce MoTion, a new pattern matcher that works on Pharo objects in the large and that focuses on flexibility and expressiveness, and then we present the syntax of both matcher tools. Finally, we compare the execution speed and the expressiveness of both pattern matchers in the context of AST matching against each other as well as against a pure Pharo request on the AST. MoTion offers a powerful base for dedicated pattern matchers and matches any kind of Pharo objects, but its genericity implies lower performances than a dedicated solution tailored for a unique usage as RBParseTreeSearcher. [9]

8.3 A Generative Approach to Modular and Versatile Virtual Machines

Garbage Collector Tuning in Pathological Allocation Pattern Applications.

Participants: Nahuel Palumbo, Sebastian Jordan Montaño, Guillermo Polito, Pablo Tesone, Stéphane Ducasse.

Automatic memory management is often supported by Garbage Collectors (GC). GC usually impacts running application performance. For tuning properly, they expose some parameters to support the adaptation of their algorithms to specific applications' scenarios. In some cases, the developers should modify the GC parameter values to achieve high performance. However, many application developers cannot be expected to perform expert analysis to determine which parameter values are the best for their application. There are techniques to find "good enough" parameter values. But, even if the overhead was reduced, it is still unknown the cause of the problem and how the GC tuning managed it. In this paper, we present a methodology to identify the causes of GC overhead in Pharo applications for tuning GC parameters. We describe the GC inside the PharoVM and its parameters, looking at how their variations change the allocation behaviour. We were able to analyse, identify and understand the GC performance issues present in one real application and suggest specific GC tuning actions. Using the suggested parameter values, we improved its performance by up to 12x and reduced GC overhead by up to 3.8x. During the experiments, we also found: 1) a bug in the production PharoVM concerning the tenuring policy, 2) a misconception about one GC parameter even for the VM developers, and 3) some possible improvements for the current GC implementation. [12]

Heap Fuzzing: Automatic Garbage Collection Testing with Expert-Guided Random Events.

Participants: Guillermo Polito, Pablo Tesone, Nahuel Palumbo, Stéphane Ducasse.

Producing robust memory manager implementations is a challenging task. Defects in garbage collection algorithms produce subtle effects that are revealed later in program execution as memory corruptions. This problem is exacerbated by the fact that garbage collection algorithms deal with low-level implementation details to be efficient. Finding, reproducing, and debugging such bugs is complex and time-consuming. In this article, we propose to fuzz heaps by generating large sequences of random heap events guided by virtual machine experts. Randomly generated events exercise the garbage collection algorithm with the objective of crashing the virtual machine and finding bugs. Once a bug is found, we

use a test case reduction algorithm to find the smaller subset of events that reproduces the issue. We implemented our approach on top of the virtual machine simulator of the Pharo Virtual Machine, to test its sequential stopthe-world generational scavenger. Experts guided our fuzzing toward the ephemeron finalization mechanism, corner allocation cases, and the heap compaction algorithm. Our prototype found 6 bugs: 3 in Pharo's ephemeron implementation which is not yet in production, 2 bugs in the default compactor which has been in production for 8 years, and 1 bug in the VM simulator used daily by VM developers. We show how such test cases were automatically reduced to trivial sequences that were easy to debug. [13]

8.4 Crosscutting all Axis / Support

Pharo DataFrame: Past, Present, and Future.

Participants: Larisa Safina, Cyril Ferlicot-Delbecque.

DataFrame is a tabular data structure for data analysis. It is a two-dimensional table (similar to a spreadsheet) with an extensive API for querying and manipulating the data. Data frames are available in many programming languages (e.g., pandas in Python or data.frame in R), they are the go-to tools for data scientists and machine learning practitioners. Pharo DataFrame was first released in 2017. Since then, the library underwent many changes and improvements. In this paper, we present the Pharo DataFrame library, show examples of its usage, and compare its API to that of pandas. We overview the changes that have been made since DataFrame v1.0, discuss the limitations of the current implementation, and present the roadmap for future. [14]

9 Bilateral contracts and grants with industry

9.1 Thales DMS, Brest, France

With the Pharo Consortium, from 2023. Industrial R&D collaboration with Dr. Eric Le Pors, lead prototyping architect at Thales DMS (Brest). We work on the Pharo core graphics library.

Participants: Pablo Tesone, Stéphane Ducasse.

9.2 Thales DMS, Brest, France

Industrial R&D collaboration with Dr. Eric Le Pors, lead prototyping architect at Thales DMS (Brest). We work on 1) unanticipated object-centric debugging of HMI prototypes 2) we study the practices of Thales with software component reuse and its impact on their development process. From 2020, ongoing.

Participants: Steven Costiou.

9.3 Berger Levrault, France

Collaboration with the software editor Berger-Levrault about software architecture remodularization. The collaboration started with an end study project exploring the architecture used in the company in order to later migrate from GWT to Angular since GWT will not be backward supported anymore in the next versions. A PhD CIFRE thesis finished in 2021. S. Bragagnolo started a CIFRE in 2020.

We now have a common team with Berger-Levrault DRIT team. We organized workshops and training sessions to improve the collaboration inside the common team. Two new PhD thesis started.

Participants: Nicolas Bortolaso, Santiago Bragagnolo, Stéphane Ducasse, Anne Etien, Nicolas Hlad, Anas Shatnawi, Benoît Verhaeghe.

9.4 CIFRE Arolla, France

We are collaborating with the council company, Arolla, about software evolution. Arolla has daily problems with identifying architecture, design, and deviations from those artefacts. The goal of Oleksandr's CIFRE (finished in 2022) experiments with different machine learning techniques that can help us automate the process of library migration. A new CIFRE PhD (from 2021) is based around the study of visualisation techniques that can help us understand legacy systems.

Participants: Nicolas Anquetil, Stéphane Ducasse, Anne Etien, Nour Jihene Agouf.

9.5 Pharo Consortium

The Pharo Consortium was founded in 2012 and is growing constantly. consortium.pharo.org (From 2012, ongoing.)

Participants: Pablo Tesone, Stéphane Ducasse, Esteban Lorenzano, Marcus Denker.

9.6 Lifeware AG, Switzerland

In collaboration with the Pharo Consortium, we improve Pharo. The goal is to be able to work with very large systems (>100K classes).

Participants: Pablo Tesone, Stéphane Ducasse, Esteban Lorenzano, Marcus Denker.

9.7 Ingenieurbüro für Bauwesen Schmidt GmbH

In collaboration with the Pharo Consortium, we improve Pharo. One focus is the use of Pharo to build user interfaces on the windows platform.

Participants: Pablo Tesone, Stéphane Ducasse, Esteban Lorenzano, Marcus Denker.

9.8 Dedalus

A collaboration started in 2021. It includes a 6 month engineer position. The goals are (1) The development of a software prototype for the identification of unused functionalities within an application developed by Inovelan. (2) Analysis of the source code of the software using the open-source software platform Moose. (3) Identification of a CIFRE thesis subject on software maintenance and development.

Participants: Nicolas Anquetil, Stéphane Ducasse, Soufyane Labsari, Anne Anne.

9.9 CIFRE Framatome, Courbevoie, France

Industrial R&D collaboration on migrating a proprietary programming language to Fortran 2003 using meta-modelisation

Participants: Nicolas Anquetil, Stéphane Ducasse, Larisa LaLarisabsari, Younoussa Sow.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Inria associate team not involved in an IIL or an international program

SADPC

Title: Systems Analyses and Debugging for Program Comprehension

Duration: 2020-2023

Coordinator: Yann-Gaël Guéhéneuc (Concordia University)

Partners:
- IDepartment of Electrical Engineering, Concordia University (Canada)
- Christopher Fuhrman: Ecole de Technologie Supérieure (Montreal)
- Fabio Petrillo, UQAC, Université du Québec à Chicoutimi
- Foutse Khomh,Polytechnique Montréal.

Inria contact: Stéphane Ducasse

Summary: Systemic changes in the past decades have pushed software systems into all aspects of our lives, from our homes to our cars to our factories. These systems, both legacy (e.g., handling contracts for the Dept of Defense of the USA since 1958) and very recent (e.g., running the latest smart factory in France in 2019), are difficult to understand by software engineers because of their intrinsic complexity. These engineers need help understanding the systems they must adapt to the new requirements of our time. The proposed associate team considers three research directions to support the software engineers maintaining and evolving large software systems: (a) system analyses and (b) debugging for (c) program comprehension. (a) Complex algorithms often act or are perceived by software engineers as black boxes because of their intrinsic and accidental complexity, both in architecture, design, and implementation. We will develop new software analyses to support algorithm understanding. (b) Previous debugging techniques assume a unique software engineer performing a solitary debugging session. We will work on a language allowing software engineers to build their own debuggers to fit their collaborative debugging strategies. (c) Previous work on program comprehension proposed views to address one single problem at a given moment of the comprehension process. They only provide a subset of the information required by software engineers. We want to propose an approach to adapt and combine views using meta-data.

Nicolas Anquetil, Stéphane Ducasse and Guillermo Polito visited the Canadian parners. We received two students from Montreal.

10.1.2 Participation in other International Programs

University of Novi Sad, Serbia
Participants: Stéphane Ducasse, Anne Etien, Nicolas Anquetil.

We collaborate with two groups of the University of Novi Sad (G. Rakic and G. Milosavljevic).

University of Zurich, Switzerland
Participants: Steven Costiou, Stéphane Ducasse, from 2020.

We collaborate with A. Bachelli on large-scale evaluations of debugging tools. This collaboration involves 3 researchers and 2 PhD students.

10.2 International research visitors

10.2.1 Visits of international scientists

Other international visits to the team

Gabriel Ullman

Status Master

Institution of origin: Concordia University, Montreal

Country: Canada

Dates: Octobert 31 to November 12

Context of the visit: SAPDC project C++ analysis

Mobility program/type of mobility: research stay

Mihajasoa Léa Fanomezana

Status PhD Sudent

Institution of origin: Université de Fianarantsoa

Country: Madagascar

Dates: September 15 to December 12

Context of the visit: Campus France scholarship

Mobility program/type of mobility: internship, research stay

Balsa Sarenac

Status PhD Student

Institution of origin: University of Novi Sad

Country: Serbia

Dates: From 1st of August to 26th to September to 26th.

Context of the visit: Refactoring engine

Mobility program/type of mobility: research stay

Marco Servetto

Status Lecturer

Institution of origin: Victoria University of Wellington

Country: New Zealand

Dates: 7 October 2023

Context of the visit: The Fearless Journey A Voyage from Pure Object-Orientation to Controlled Mutability

Mobility program/type of mobility: visit

Imen Sayar

Status Postdoc at Smart Team

Institution of origin: Université de Toulouse 2 Jean-Jaurès

Country: France

Dates: March 6 – March 10, 2023

Context of the visit: Exchange with team researchers to prepare for "associate professor" position application at University of Lille

Mobility program/type of mobility: research stay

Leonel Merino

Status Associate Prof

Institution of origin: Pontifical Catholic University of Chile

Country: Chile

Dates: December 18 to 22 December 2023

Context of the visit: Analysis development integrated environment

Mobility program/type of mobility: research stay

Aurel Lucrich Ikama-honey

Status PhD Student

Institution of origin: Ecole polytechnique Montreal

Country: Canada

Dates: July 3rd to September 8th

Context of the visit: ML analysis

Mobility program/type of mobility: research stay

10.2.2 Visits to international teams

Research stays abroad

Nicolas Anquetil

Visited institution: École de Technologie Supérieure, Montréal, Québec

Country: Canada

Dates: 18/07 – 30/07

Context of the visit: Project SAPDC, meeting with colleagues and practical sessions

Mobility program/type of mobility: research stay

Stéphane Ducasse

Visited institution: ETS, UQAM, Concordia

Country: Canada

Dates: 10 days October 2023

Context of the visit: SADPC International team

Mobility program/type of mobility: research stay

Guillermo Polito

Visited institution: ETS, UQAM, Concordia

Country: Canada

Dates: 10 days October 2023

Context of the visit: SADPC International team

Mobility program/type of mobility: research stay

10.3 European initiatives

10.3.1 H2020 projects

EVREF is part of the COST project CERCIRAS: Connecting Education and Research Communities for an Innovative Resource Aware Society. www.cost.eu/actions/CA19135

10.4 National initiatives

SWHSec: Leveraging Software Heritage to Enhance Cybersecurity
Partners: Evref, Software Heritage (From 2023 to 2025).

Participants: Cyril Ferlicot-Delbecque, Stéphane Ducasse, Imen Sayar, Anne Etien, Nicolas Anquetil.

The rise of Open Source has accelerated innovation by allowing massive reuse of a huge number of freely available software components developed by a vast community distributed around the world. This has had serious consequences on the software supply chain, with the introduction of a large number of dependencies on components whose quality level is difficult to assess and control: they can contain vulnerabilities, and become sources of attacks on the systems that depend on them, as we saw with the Log4J incident. Recent examples of deliberately sabotaged software in response to the invasion of Ukraine have shown how the line between well-intentioned and malicious actors in the software development world is becoming increasingly blurred.

The urgency of addressing these issues is now clearly perceived, as seen for example in the May 2021 White House Executive Order, which explicitly mentions the need to "ensure and attest, to the extent possible, the integrity and provenance of open source software."

To meet this imperative, it is necessary to be able to analyze the millions of publicly available software projects, study their development history, and extract relevant information.

We are fortunate to have the Software Heritage archive, an initiative launched about 6 years ago by Inria in partnership with UNESCO, which already contains more than 12 billion unique source files from more than 180 million different origins, with all their development history.

This project brings together a group of research teams with significant expertise in software source code analysis to leverage the unprecedented resource that is the Software Heritage archive and explore the possibilities it opens up in terms of cybersecurity. New features needed to enrich the archive with security-relevant information such as component dependencies and links to known vulnerabilities will be developed, used to trace the origin and impact of vulnerabilities, and automatic detection and remediation from the patterns thus detected will be explored.
These developments will provide the basis for making Software Heritage effectively usable in industrial and cyber defense applications.

ANR JCJC Sapper
Partners: EVREF, Sigma, UQAM (Quebec) (From 2023 to 2027).

Participants: Guillermo Polito, Pablo Tesone, Jean Privat, Remi Bardenet.

In Sapper we propose a holistic approach to reduce the cost of benchmarking. Namely, we will study how to build relevant, reproducible, and interpretable benchmark programs. We will automate the generation, selection, execution and interpretation of benchmarks by reuniting fundamental, practical, and empirical knowledge from programming language implementation, software engineering, and statistics.

ARCAD, Lab-STICC, Brest, France

Participants: Guillermo Polito, Pablo Tesone, Stéphane Ducasse.

We collaborate since the beginning of 2021 with the ARCAD team of the Lab-STICC in Bretagne (Prof L. Lagadec). We started at the beginning of the year with a common workshop between the two teams looking for collaboration points. G. Polito and P. Tesone are now collaborating with the PhD of Q. Ducasse on Just-In-Time compiler technology for extensible ISA processors such as RISC-V.

École Nationale d'Ingénieurs de Tarbes

Participants: Marcus Denker.

With Cédrick Béler (ENIT/LGP/ICE) we are exploring the life-cycle (contextual time relation) of data, information, and knowledge in the context of Object-Oriented data models.

ANR JCJC OCRE
Partners: EREF, SmArtSE (UCAQ, Quebec), UX Prototyping (Thales DMS, Brest) (From 2022 to 2024).

Participants: Steven Costiou, Valentin Bourcier, Marcus Denker.

The objectives of the OCRE project are to study the fundamental and practical limits that hinder the implementation, the evaluation, and the adoption of object-centric debugging. We propose to build the first generation of object-centric debuggers, in order to identify and evaluate its real benefits to OOP debugging. We argue that these debuggers have the potential to drastically lower the cost (time and effort) of tracking and understanding hard bugs in OOP.

Action Exploratoire Inria: AlaMVic

Participants: Guillermo Polito, Pablo Tesone, Nahuel Palumbo.

Summary: Language Virtual Machines (VMs) are pervasive in every laptop, server, and smartphone. Industry-level VMs use highly-engineered optimization techniques, often handcrafted by experts, difficult to reproduce, replicate and change. Such optimization techniques target mostly speed improvements and are incompatible with constraints such as space and energy efficiency important in the fields of IoT or robotics. In AlaMVic1 we propose to approach VM construction using a holistic generative approach, in contrast with existing approaches that focus on speed and single VM components such as the JIT compiler. We explore how to transform handcrafted optimizations into generation heuristics, how they are applied and combined in fields such as IoT and robotics, and new methods and metrics to evaluate VMs in such fields.

10.5 Regional initiatives

IMT Douai
Collaboration with Prof L. Fabresse and Prof. N. Bouraqadi. The PhDs of P. Tesone, P. Misse, and C. Hernandez are joint PhD with the team of IMT Douai.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

- ESUG 2023 Lyon, France (90 PP, 4 Days)
- Larisa Safina: 16th Interaction and Concurrency Experience (ICE) workshop co-located with 18th International Federated Conference on Distributed Computing Techniques (DisCoTec)
- Larisa Safina: Agility with Microservices Programming workshop colocated with 17th European Conference on Software Architecture (ECSA)

General chair, scientific chair

- Stéphane Ducasse: ESUG 2023 Lyon, France

Member of the organizing committees

- Steven Costiou: Journées Nationales du GDR-GPL 2023 Organisateur et Session Chair des sessions du GT Debugging
- Marcus Denker, Pablo Tessone: ESUG 2023

11.1.2 Scientific events: selection

Chair of conference program committees Stéphane Ducasse: IWST 2023 Lyon, France

Member of the conference program committees

- Anne Etien: DEBT 2023, VISSOFT 2023, Benevol 2023, SANER (RENE track) 2024, ICPC 2024, ICSME 2024.
- Guillermo Polito PC of MPLR'23
- Marcus Denker: RAW23 Workshop
- Steven Costiou: First Workshop on Future Debugging Techniques: DEBT 2023

Larisa Safina

- 4nd IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)
- European Conference on Service-Oriented and Cloud Computing (ESOCC)
- The International Conference on Microservices, Pisa, Italy
- International Workshop on Smalltalk Technologies ESUG: IWST
- International conference on formal methods in software engineering (FormaliSE) colocated with 45rd International Conference on Software Engineering (ICSE2023) (Artefact Evaluation)
- 25rd International Conference on Coordination Models and Languages (COORDINATION) (Artefact Evaluation)
- 44rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) (Artefact Evaluation)

Reviewer

- Steven Costiou: SANER 2023
- Imen Sayar: subreviewer for the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2024)
- Imen Sayar: subreviewer for the 22nd Belgium-Netherlands Software Evolution Workshop (BENEVOL 2023)

11.1.3 Journal

Member of the editorial boards

- Larisa Safina: Special Issues of the Journal of Logical and Algebraic Methods in Programming (International Journal Guest Editor)

Reviewer - reviewing activities

- Anne Etien: Journal of Systems and Software, Science of Computer Programming
- Guillermo Polito: COLA journal
- Larisa Safina: ACM Transactions on Autonomous and Adaptive Systems
- Steven Costiou: IEEE Software

11.1.4 Invited talks

- Guillermo Polito — Talk ETS Montréal, October 10
- Guillermo Polito — Séminaire LATECE UQAM Montréal, October 11
- Guillermo Polito — Talk Polytechnique Montréal, October 16
- Stéphane Ducasse — Split Summer School

11.1.5 Leadership within the scientific community

GT GLIA working group of the CNRS GDR GPL
Anne Etien is co-leader of the GT GLIA working group of the CNRS GDR-GPL (Software Engineering and artificial intelligence) from 2020 till June 2023

GT Debugging working group of the CNRS GDR GPL
Steven Costiou is leader of the GT Debugging working group of the CNRS GDR GPL. This working group aims to gather any researcher, engineer or GDR team interested in software debugging problems. (from 2020)

Committees

- Steven Costiou: Prix de thèse du GDR-GPL 2023 Program Committee
- Larisa Safina: Microservices Community Ethics Committee

11.1.6 Scientific expertise

Anne Etien: Evaluation ANR

11.1.7 Research administration

- Anne Etien animates the thematic group of Software Engineering and is member of scientific council of CRIStAL lab.
- Anne Etien: Directrice des Études de la L3 MIAGE, Université de Lille
- Anne Etien: Elected as a Member of CNU (section 27).

11.2 Teaching - Supervision - Juries

- Master: Steven Costiou, Fondamentaux du debugging, Université de Lille, 4hCM 8hTD
- Master: Steven Costiou, Debugging: bases and implementation, Université de Brest, 12hCM 12hTD
- Master: Steven Costiou, Conception et modélisation objet, Polytech Lille, 10hCM 10hTD
- Master: Stéphane Ducasse, Meta, Université de Lille, 12hTD
- Master: Stéphane Ducasse, Conception avancée, Université de Lille, 60hTD
- Licence: Anne Etien, Introduction à la programmation, 40h, L1, Université de Lille
- Licence: Anne Etien, projet, 24h, L2, Université de Lille
- Licence: Anne Etien, Conception orientée objet, 18, L3, Université de Lille
- Licence: Anne Etien, Bases de données relationnelles, 16h, L3, Université de Lille
- Licence: Anne Etien, Génie Logiciel, 27h, L3, Université de Lille

- Master: Anne Etien, Metamodelisation, 30h, M2, Université de Lille
- Master: Guillermo Polito, Analyse et Verification de Logiciel, Université de Lille, 16h CM
- Master: Guillermo Polito, Conception et Paradigmes de Programmations par la Pratique, Université de Lille, 48hTD
- Licence: Guillermo Polito, Meta, Université de Lille, 13hTD
- Master: Guillermo Polito, Compiler Fuzzing, Universidad de Buenos Aires, 15h CM
- Licence: Cyril Ferlicot-Delbecque, Génie Logiciel, 18h, L3, Université de Lille
- Master: Imen Sayar, Introduction à la Sécurité Informatique (ISI), FST Université de Lille, 18h TD
- Licence: Imen Sayar, Génie logiciel (GL), FST Université de Lille, 18h TD
- Licence: Imen Sayar, Programmation des systèmes (PDS), FST Université de Lille, 21h TD
- Licence: Imen Sayar, Conception orientée objet (COO), FST Université de Lille, 36 TD
- Licence: Imen Sayar, Projet, FST Université de Lille, 48h TD
- Licence: Imen Sayar, Bases de données 1 (BDD1), FST Université de Lille, 21h TD
- Licence: Iona Thomas, Génie Logiciel, 18h, L3, Université de Lille
- Licence: Iona Thomas, Introduction à la programmation, 18h, L1, Université de Lille
- Licence: Iona Thomas, Meta, 12h, L3, Université de Lille
- Master: Gabriel Darbord, Conception avancée, Université de Lille, 18hTD
- Licence: Aless Hosry, Génie logiciel (GL), FST Université de Lille, 18h TD
- Licence: Aless Hosry, Algorithmes et programmation (AP), FST Université de Lille, 36h TD
- Licence: Larisa Safina, Meta, Université de Lille, 18hTD
- Licence: Larisa Safina, projet, 24h, L2, Université de Lille
- Master: Marcus Denker, 2 hours, Advanced Reflection, VUB Brussels, Belgium.

11.2.1 Supervision

- Santiago Bragagnolo, Migration de programmes légataires vers des architectures Web: le cas de de la migration de programmes Microsoft Access vers Angular / Microservices, CIFRE Berger-Levrault, Stéphane Ducasse, Nicolas Anquetil. May 17, 2023. [19]
- Nour-Dijene Agouf, Visualisations for Real software systems, since 2021, CIFRE Arolla, Stéphane Ducasse, Anne Etien, 12 December 2023. [18]
- Maximilian Ignacio Willembrinck Santander, Scriptable Time-Traveling Debuggers, Inria, Anne Etien, Steven Costiou, 21 November 2023 [20]
- PhD in progress: Iona Thomas, Elements of Language Strenghtening, since 2021 Stéphane Ducasse, Pablo Tesone, Guillermo Polito
- PhD in progress: Aless Hosry, Model transformations for automatic source code modification, Nicolas Anquetil.
- PhD in progress: Gabriel Darbord, Automatic Test Generation, since october 2022, Inria through the EPC with BL, Anne Etien, Nicolas Anquetil.

- PhD in progress: Valentin Bourcier, Reducing the cost of debugging with the first generation of object-centric debuggers, since october 2022, Inria, Steven Costiou
- PhD in progress: Nahuel Palumbo, Virtual Machine Generation Techniques, since november 2022, Inria, Stéphane Ducasse, Guillermo Polito.
- PhD in Progress: Soufyane Labsari, DSL et cartes scriptables pour la cartographie de systèmes patrimoniaux, since December 2023, Inria, Anne Etien and Nicolas Anquetil

11.2.2 Juries

- Nicolas Anquetil: Céline Deknop, *Understanding large codebase refactoring through differencing*, Louvain School of Engineering, Université catholique de Louvain, October 13, 2023.
- Anne Etien: Mikael Salson, HDR, "Méthodes sans alignement et indexation pour l'analyse de données nucléiques massives" novembre 2023, à l'Université de Lille, as president.
- Anne Etien: Elena Kornyshova, HDR, "Context- and Intention-based Configuration in the Era of Digitalisation" novembre 2023, à l'Université Paris I, as reviewer.

11.3 Popularization

11.3.1 Internal or external Inria responsibilities

- Anne Etien is elected member of the center committee of Inria Lille Nord Europe center.
- Anne Etien is elected member of the Computer Science Department council of University of Lille
- Anne Etien is member of the Scientific council of CRIStAL
- Marcus Denker was a member of the AGOS board (Section culture) of Inria Lille
- Guillermo Polito is a member of the Argentinian Uqbar Foundation
- Guillermo Polito, Stéphane Ducasse and Marcus Denker are members of the Pharo Board

11.3.2 Articles and contents

- Multiple posts on the Pharo DevBlog
- The Book *Testing in Pharo* was published [17]

11.3.3 Education

The Advanced Object-Oriented Design and Development with Pharo Mooc was released.

11.3.4 Interventions

- We organized Public Pharo Sprints every last Friday of the month
- Iona Thomas: "Science en Livre" event at Lilliade for schools
- Iona Thomas: "Fête de la science" at Inria for (niveau collège)
- Iona Thomas: "Forum académique de la culture scientifique" - 3h
- Iona Thomas: "13h45 presentation" at Inria - internal scientific mediation event
- Iona Thomas: Meeting with researchers event as part of "L Décodent l'@venir", a collective observation internship (3ème & Terminale)
- Organization of the Pharo Summer School at Split (Sept 2023)

12 Scientific production

12.1 Major publications

[1] G. Polito, P. Tesone, J. Privat, N. Palumbo and S. Ducasse. 'Heap Fuzzing: Automatic Garbage Collection Testing with Expert-Guided Random Events'. In: ICST 2023 - International Conference on Software Testing. Dublin, Ireland, 16th Apr. 2023. URL: https://inria.hal.science/hal-03962007.

12.2 Publications of the year

International journals

[2] G. Darbord, B. Verhaeghe, A. Etien, N. Anquetil, A. Shatnawi, A. Seriai and M. Derras. 'Migrating the Communication Protocol of Client-Server Applications'. In: *IEEE Software* 40.4 (July 2023), pp. 11–18. DOI: 10.1109/MS.2023.3263019. URL: https://hal.science/hal-04050310.

[3] I. Thomas, V. Aranega, S. Ducasse, G. Polito and P. Tesone. 'A VM-Agnostic and Backwards Compatible Protected Modifier for Dynamically-Typed Languages'. In: *The Art, Science, and Engineering of Programming* (15th June 2023). DOI: 10.22152/programming-journal.org/2024/8/2. URL: https://hal.science/hal-04119017.

International peer-reviewed conferences

[4] N. J. Agouf, S. Labsari, S. Ducasse, A. Etien and N. Anquetil. 'A Visualization for Client-Server Architecture Assessment'. In: IEEE Working Conference on Software Visualization. Bogota, Colombia, 1st Oct. 2023. URL: https://hal.science/hal-04231797.

[5] S. Bliudze, P. van den Bos, M. Huisman, R. Rubbens and L. Safina. 'JavaBIP meets VerCors: Towards the Safety of Concurrent Software Systems in Java'. In: FASE 2023 - 26th International Conference on Fundamental Approaches to Software Engineering. Vol. 13991. Lecture Notes in Computer Science. Paris, France: Springer Nature Switzerland, 20th Apr. 2023, pp. 143–150. DOI: 10.1007/978-3-031-30826-0_8. URL: https://inria.hal.science/hal-03911393.

[6] G. Darbord, A. Etien, N. Anquetil, B. Verhaeghe and M. Derras. 'A Unit Test Metamodel for Test Generation'. In: *CEUR Workshop Proceedings*. International Workshop on Smalltalk Technologies. Lyon, France, 2023. URL: https://hal.science/hal-04219649.

[7] G. Galindo-Gutierrez, N. Maxilimiliano, B. Alison Fernandez, N. Anquetil and A. Juan Pablo Sandoval. 'A manual categorization of new quality issues on automatically-generated tests'. In: Proceedings of the 39st IEEE International Conference on Software Maintenance and Evolution (ICSME'23). Bogota, Colombia, 1st Oct. 2023. DOI: 10.1109/ICSME58846.2023.00035. URL: https://hal.science/hal-04344531.

[8] A. Hosry and N. Anquetil. 'External dependencies in software development'. In: *External dependencies in software development*. Quality of Information and Communications Technology, 16th International Conference, QUATIC 2023. Vol. 1871. Springer CCIS Series (Communications in Computer and Information Science). Aveiro (Portugal), Portugal: Springer, 2023, pp. 215–232. URL: https://hal.science/hal-04217300.

[9] A. Hosry, V. Aranega, N. Anquetil and S. Ducasse. 'Pattern matching in Pharo'. In: *CEUR Workshop Proceedings*. IWST 2023 - International Workshop on Smalltalk Technologies. Lyon, France, 2023. URL: https://hal.science/hal-04217930.

[10] S. J. Montaño, N. Palumbo, G. Polito, S. Ducasse and P. Tesone. 'Improving Performance Through Object Lifetime Profiling: the DataFrame Case'. In: IWST 2023 - International Workshop on Smalltalk Technologies. Lyon, France, 28th Aug. 2023. URL: https://hal.science/hal-04253865.

[11] T. Oda, G. Dur, S. Ducasse and H. D. Macedo. 'Implementation-First Approach of Developing Formal Semantics of a Simulation Language in VDM-SL'. In: 21st Overture Workshop. Lubeck, Germany, 10th Mar. 2023. URL: https://inria.hal.science/hal-04030293.

[12] N. Palumbo, S. J. Montaño, G. Polito, P. Tesone and S. Ducasse. 'Garbage Collector Tuning in Pathological Allocation Pattern Applications'. In: *CEUR Workshop Proceedings*. IWST 2023 - International Workshop on Smalltalk Technologies. Lyon, France, 29th Aug. 2023. URL: https://inria.hal.science/hal-04225588.

[13] G. Polito, P. Tesone, J. Privat, N. Palumbo and S. Ducasse. 'Heap Fuzzing: Automatic Garbage Collection Testing with Expert-Guided Random Events'. In: ICST 2023 - International Conference on Software Testing. Dublin, Ireland, 16th Apr. 2023. URL: https://inria.hal.science/hal-03962007.

[14] L. Safina, O. Zaitsev, C. Ferlicot-Delbecque and P. I. Sow. 'Pharo DataFrame: Past, Present, and Future'. In: International Workshop on Smalltalk Technologies. Lyon, France, 29th Aug. 2023. URL: https://hal.science/hal-04212030.

[15] I. Thomas, S. Ducasse, P. Tesone and G. Polito. 'Pharo: a reflective language - A first systematic analysis of reflective APIs'. In: *CEUR-WS Proceedings*. IWST 23 — International Workshop on Smalltalk Technologies. Lyon, France, 29th Aug. 2023. URL: https://inria.hal.science/hal-04217271.

[16] A. Vanègue, V. Bourcier, F. Petrillo and S. Costiou. 'Debugging Video Games: A Systematic Mapping'. In: DEBT 2023 - First Workshop on Future Debugging Techniques. Seattle, United States: ACM, 17th July 2023, pp. 23–30. DOI: 10.1145/3605155.3605865. URL: https://inria.hal.science/hal-04139070.

Scientific books

[17] S. Ducasse, G. Polito and J. P. Sandoval Alcocer. *Testing in Pharo: Testing in Pharo*. 2023. URL: https://inria.hal.science/hal-04216172.

Doctoral dissertations and habilitation theses

[18] N. J. Agouf. 'Bird-Eye Views of Object Oriented Software'. Lille University, 12th Dec. 2023. URL: https://hal.science/tel-04392323.

[19] S. Bragagnolo. 'A Holistic Approach to Migrate Industrial Legacy Systems'. Universite de Lille; Inria; Universite de Lille, 17th May 2023. URL: https://inria.hal.science/tel-04132315.

[20] W. S. Maximilian Ignacio. 'An Interactive Debugging Approach Based on Time-traveling Queries'. Inria, 21st Nov. 2023. URL: https://theses.hal.science/tel-04398079.

Reports & preprints

[21] S. Bragagnolo, N. Anquetil, S. Ducasse, A.-D. Seriai and M. Derras. *Interactive, Iterative, Tooled, Rule-Based Migration of Microsoft Access to Web Technologies*. 17th May 2023. URL: https://inria.hal.science/hal-04181591.

[22] S. J. Montaño, G. Polito, S. Ducasse and P. Tesone. *ILLIMANI Memory Profiler -A Technical Report*. INRIA Lille - Nord Europe, 6th Mar. 2023. URL: https://hal.science/hal-04225251.

[23] A. Shatnawi, B. Rima, Z. Alshara, G. Darbord, A.-D. Seriai and C. Bortolaso. *Telemetry of Legacy Web Applications: An Industrial Case Study*. 20th Oct. 2023. DOI: 10.36227/techrxiv.24449092. URL: https://hal.science/hal-04344518.

[24] Y. Sow, L. Safina, L. Brault, I. Papa, S. Ducasse and N. Anquetil. *Parsing Fortran-77 with proprietary extensions*. 2023. URL: https://hal.science/hal-04205262.

[25] I. Thomas, S. Ducasse, P. Tesone and G. Polito. *A classification of runtime reflective operations in Pharo*. Inria Lille - Nord Europe, CRIStAL - Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189, 3rd Oct. 2023. URL: https://inria.hal.science/hal-04225720.

Other scientific publications

[26] A. Vanègue and S. Costiou. 'An extensible production-level debugger'. In: Journées Nationales du GDR-GPL 2023. Rennes, France, 5th June 2023. URL: https://inria.hal.science/hal-04130163.

12.3 Cited publications

[27] J. P. Sandoval Alcocer, A. Bergel, S. Ducasse and M. Denker. 'Performance Evolution Blueprint: Understanding the Impact of Software Evolution on Performance'. In: *VISSOFT - 1st IEEE Working Conference on Software Visualization*. Ed. by A. C. Telea. Eindhoven, Netherlands: IEEE, Sept. 2013, pp. 1–9. DOI: 10.1109/VISSOFT.2013.6650523. URL: https://inria.hal.science/hal-00849004.