PLOMO2 Associate Team Final Report

A Bergel, Stéphane Ducasse, Marcus Denker, J Fabry

PLOMO2 Associate Team final report

A. Bergel S. Ducasse M. Denker J. Fabry

October 27, 2016

Associate Team acronym: PLOMO2

Period of activity: 2014-2016

Principal investigator (Inria): S. Ducasse – RMoD – Lille Nord Europe

Principal investigator (partner): A. Bergel – DCC – Pleiad – University of Chile.

Other participants: M. Denker (RMoD), J. Fabry (DCC), D. Pollet (RMoD), R. Robbes (DCC).

Project Overview

Performing effective software development and maintenance are best achieved with effective tool support. Provided by a variety of tools, each one presenting a specific kind of information supporting the task at hand. The goal of the first PLOMO was to develop new meta tools to improve and bring synergy in the existing infrastructure of Pharo (for software development) and the Moose software analysis platform (for maintenance). With PLOMO2, we build on top of this work and invent a new generation of tools to navigate and profile programs. The work was carried out in three work packages:

- Work Package One: Reflective Foundations

- Work Package Two: Visualizations Engines

- Work Package Three: Tools for Analyzing Dynamic Information
  (combining work package one and two).

1 Original Project Plan

1.1 Work Package One

For both for program profiling and programmer activity recording, we need a way to deeply hook into the system. We plan for this to improve the reflective infrastructure of Pharo.

During the previous period we worked on OPAL, a new compiler toolchain for Pharo. In essence the work on the compiler framework does not move forward the state of the art in compiler technology. However it has already been proven to be a crucial building block. It provided the fundamentals for many research experiments and new features for Pharo. For example, Pleiad and RMoD collaborated on a novel type system for Pharo that was realized using the new compiler infrastructure. Other examples are the many new features of Pharo 3 that use parts of the new compiler chain, like AST-based navigation in the editor or breakpoints.

The Opal flexible compiler infrastructure will allow us to build a new generation of reflective infrastructure. Such infrastructure is keys to support tools (profilers) and new language design (AOP, proxies, isolation). We are now in a position to perform a new iteration on the reflective layer and metaobject protocol for Pharo. In 2005 we started to develop new reflective foundations and we learned a lot (partial reflection, metalevel recursion handling, object-centric on the fly propagation, proxies, isolation). We are now in a situation to step back and redesign the reflective layer based on all the cases we identified and developed.

The results of this work package will be used and thus validated by WP3. In addition, we plan to realize the next generation of the AOP framework Phantom using the reflection framework. This especially makes sure that the framework is general and not just limited for the use in tools.

1.2 Work Package Two

Plomo made a lot of progress by realizing the Roassal visualisation framework. We plan to enhance the infrastructure to visually render software-related data. We will investigate two complementary directions:

1. Interaction and scalability in Roassal 2D. Roassal is an Agile Visualization Engine that offers large range of options to make the user interact with the visualization. Visualization software require particular customizations to cope with large amount of data. Navigation should support interactive and semantic zooming, which is an essential feature for navigating large amount of data elements. Additional layers, including drawing dependency matrices, vectorial and interactive charts are required

2. Visual representation of 3D environment gives a sensation of immersion that is often key to identify some patterns in a data structure. Current programming environments exploiting 3D are either fixed to a particular representation or unfit to be used as the base of a programming environment

1.3 Work Package Three

The goal of the third work package is to build on top of both WP1 and WP2: Combining dynamic information with visualization to improve the development environment.

1. Visualization to navigate and browse source code – System complexity is a polymetric view efficient at representing class hierarchies in a software source code. It has been shown that system complexity is effective for software maintenance. We will augment the range of application of it by using it to navigate and browser application source code.

2. Visualizing dynamic information to navigate source code. The static view of code should be enhanced with dynamic information gathered from program execution. A good example here is visualization of profiling and tracing information.

3. Context based on history – It has been shown that analyzing past user actions may precisely infer what will be the next user actions. Unfortunately, history is poorly supported in traditional programming environment. Making history a first class element in the programming environment is the topic of this work package.

This work package will validate both WP1 and WP2.

2   Results

2.1   Work Package One Achievements: Reflective Foundations

Progress on a new iteration on the reflective layer and metaobject protocol for Pharo: We have implemented the Slots Mechanism in Pharo, which resulted in a first-class reification of instance variables and their accessing through assignment and reading. This allows us to hook into reading and writing of instance variables and perform meta-level operations on them, *e.g.,* adding breakpoints to the reading of a specific instance var, or a transparent notification mechanism for when a specific instance variable is written. The latter can be used, *e.g.,* in WP3 to dynamically update a visualization of the object that contains the instance variable whenever the variable changes.

Reflective operations in the form of meta-links can be added to any expression tree. All variables can now be annotated with meta data. Reflectivity has been improved to allow Meta-Links to be put on First Class Variables. All work has been integrated in Pharo5 development and is used already by other projects of RMOD and a research project of the University of Lugano/Switzerland. The stay of Marcus Denker in Chile in November 2015 had the goal to make the new reflective infrastructure to be reliable enough for other PLOMO sub projects to use (namely LRP and Profilers). A breakpoint system based on Metalinks has been added to Pharo5 and is being improved for the release of Pharo6.

The object layout functionality of Pharo allowed us to work on supporting C-Types. The Opal Compiler was extended with primitive types and to emit new low-level bytecode instructions. Combined, this allows performance for use-cases like 2D and 3D frameworks to be speed up by 50-400% [**13**].

2.2   Work Package Two Achievements: Visualizations Engines

Two complementaries axes were investigated. Roassal 2D is a visualization engine to render data. Roassal 2D is employed to visualize any arbitrary set of data, including software-related data [**5, 9**] (*e.g.,* source code, module dependencies), numerical set of numbers [**6**] (*e.g.,* metric values). Roassal 2D has been improved from the interaction and scalability point of view. Roassal 2D is regularly used to visualize software systems large of thousands of classes.

The second axis is the visualization in 3D. For that purpose, we have improved the FFI support of Pharo to use external libraries. With that, we created binding to SDL2 which allowed to move the window management code from the Virtual Machine to Pharo (OSWindow). The 2D infrastructure work (OSWindow) has been taken up by another Project (Inria in collaboration with Thales). It has been extended with multi-touch support and a gesture engine. Besides pre-defined low level gestures (*e.g.,* pinch or multi-finger swipe), the engine is extensible with user-defined gestures. The Woden 3D graphics engine is a first demo that uses this foundation

to bring the ideas of Roassal to 3D. The new next generation graphics frameworks like Metal and Vulcan where the driving force to revisit Woden. The new framework Woden 2 uses the lowcode subsystem (see WP1) for a new generation, high perfomance 3D framework for Pharo. This will be the foundation of Roassal3D soon. Woden has already seen use outside of the PLOMO project, for example it is used for experiments with virtual reality at a large french multi-national company.

2.3 Work Package Three Achievements: Tools for Analyzing Dynamic Information

Live Robot Programming (LRP) [10] as a case for the first two points of WP3: Visualization to navigate and browse source code, Visualizing dynamic information to navigate source code. Firstly, in LRP the (Domain-Specific) source code is shown in a (Domain-Specific) visualization made in Roassal2 and Spec. Clicking on the entities in the visual representation causes a navigation action in the source code of the program. Secondly, in the LRP visualization the dynamic information of the state of variables in scope is shown, as well as which piece of the code is currently running, and this is partially enabled by Slots. The visit of Marcus Denker to Chile in spring 2015 helped to make serious progress on the base implementation infrastructure: The LRP prototype now is based in the First Class Variables (Slots and Globals) that are available in Pharo.

We worked on refining the user interface framework Spec [4,16], together with its integration with the visualization of dynamic information using Roassal as well as the GT Inspector framework for the analysis of live objects. The goal is to allow a developer to build his own custom user interface for the visualization of dynamic information, easily combining classical UI widgets with more advanced visualizations (provided by Roassal and/or the powerful object inspection and modification abilities of the GT Inspector). One example of a specific UI built this way is the work of Fabry and Sinclair [14] allowing the analysis of the dynamic evolution of simulated object trajectories in different physics engines *e.g.,* the Siconos engine by INRIA Bipop. This work on Spec is enhanced by the writing of extensive documentation on the framework, which was shown to be necessary for developers to effectively be able to construct such user interfaces.

2.4 Work outside of work packages

Close proximity among the researchers also allowed work outside of the work packages to be started. One example is a preliminary effort in speeding up collections handling [12], another the work on identifying classes in legacy JavaScript code [15].

3 Production

3.1 Software

- Pharo: Pharo3 has been released Spring 2014, `http://pharo.org`

- Pharo: Pharo4 has been released Spring 2015.

- OSWindow has been integrated in Pharo4.

- Pharo: Pharo5 has been released Spring 2016, Pharo6 has seen active development with a release planned for early 2017.

- Reflectivity has been integrated in Pharo5

- Moose: Version 5.0 was released. `http://moosetechnology.org`

- Moose: Version 6.0 was released.

- Roassal2D: Fully based on Athens, the vector graphics support in Pharo

- Roassal3D: first version of a 3D version of Roassal. `https://vimeo.com/141916068`

- Woden: a 3D OpenGL graphic engine, the base of Roassal3D.

- Woden 2: a new 3D graphic engine for the next generation 3D frameworks (Metal3D, Vulcan3D), it is the base of the next generation of Roassal3D.

    - Talk at ESUG 2016: `http://goo.gl/RuW36E`

- The lowcode subsystem has been realised
  `http://goo.gl/jHjndl`

- Improvement of FFI (Foreign Function Interface). Work on UFFI to use Lowcode.

- The Opal Compiler has been further improved. It has been extended with support for Slots / Layouts and is used as the basis for Lowcode.

4 Overview of the Activities

The activities of the PLOMO2 members were mainly oriented towards scientific publications as well as software development.

4.1 Organization

- The IWST Workshop series at ESUG in 2014, 2015, 2016 (`http://esug.org`).

- Coding sprints coordinated at the same time both in Lille and Santiago over the whole period.

4.2 Research Visits

PLOMO enables the participants RMoD (INRIA Lille Nord Europe) and Pleiad (DCC University of Chile) to collaborate by funding their trips.
2014

- Ronie Saldago: 24/08/2014 until 07/09/2014 to Lille.
  Subject was FFI and OSWindow.

- Miguel Campusano: 16/08/2014 until 11/09/2014 to Lille.
  Subject was Slots and visual representation of code.

- Alexandre Bergel: 13/12/2014 until 01/01/2015 to Lille.
  Subject: system support for advanced profiling.

- Juraj Kubelka: 06/12/2014 until 19/12/2014 to Lille.
  First visit to RMoD to plan future collaboration.

2015

- Johan Fabry: January 2015 to Lille for Pharo Days Lille, PLEIAD funded.

- Alexandre Bergel: 01/07/2015 until 27/07/2015 to Lille.

- Johan Fabry: July 2015 to Lille for ESUG, PLEIAD fundet.

- Pierre Chanson: July 2015 and September 2015 to Lille (to discuss PhD with RMoD)

- Miguel Campusano: 20/09/2015-07/10/2015, to Lille.

- Alexandre Bergel: Dec 2015, to Lille .

- Marcus Denker: 2/1/2015 until 20/1/2015, to Santiago, sponsored by PLEIAD.
  Subject is Slots.

- Marcus Denker: Nov 2015 payed by PLEIAD, to Santiago.
  Subject was Reflection Framework.

2016

- Ronie Saldago: April - September 2016, to Lille.
  Working on a new FFI library and a 3D visualization engine

- Alexandre Bergel: December 2016, to Lille.
  Working on a new platform for software and data visualization

- Johan Fabry: July 2016, to Lille.
  Subject was the writing of the book on Spec.

- Marcus Denker: December 2016, to Santiago.
  Working on Reflective foundations. Flight payed by PLEAID.

4.3   PLOMO Associated publications

[1] Johan Fabry, Andy Kellens, Simon Denier, Stephane Ducasse. "AspectMaps: Extending
    Moose to Visualize AOP Software". Science of Computer Programming, volume 79, number 1, pp.6-22, January 2014, Elsevier
    `https://hal.inria.fr/hal-01086997`

[2] Johan Fabry, Romain Robbes, Marcus Denker. "DIE: A Domain Specific Aspect Language
    for IDE Events". Journal of Universal Computer Science, volume 20, number 2, pp.135-
    168, February 2014
    `https://hal.inria.fr/hal-00936376`

[3] Esteban Allende, Oscar Callau, Johan Fabry, Éric Tanter, Marcus Denker. "Gradual
    Typing for Smalltalk". Science of Computer Programming, volume 96, number 1, pp.52-
    69, December 2014, Elsevier
    `https://hal.inria.fr/hal-00862815`

[4] Benjamin Van Ryseghem, Stephane Ducasse, Johan Fabry. "Seamless Composition and
    Reuse of Customizable User Interfaces with Spec". Science of Computer Programming,
    volume 96, number 1, pp.34-51, December 2014, Elsevier
    `https://hal.inria.fr/hal-00915350`

[5] Alexandre Bergel, Sergio Maass, Stephane Ducasse, Tudor Girba. "A Domain-Specific Language For Visualizing Software Dependencies as a Graph". Proceedings of 2nd IEEE Working Conference on Software Visualization (VISSOFT NIER)
https://hal.inria.fr/hal-01369700

[6] Popular Article LinuxMag: Roassal and the Charter charting framwork.
http://www.gnulinuxmag.com

[7] Miguel Campusano and Johan Fabry. "From Robots to Humans: Visualizations for Robot Sensor Data." In the third IEEE Working Conference on Software Visualization (VISSOFT 2015), Sep 27-28, 2015 , IEEE press.
http://dx.doi.org/10.1109/VISSOFT.2015.7332424

[8] Damien Cassou, Stephane Ducasse, Luc Fabresse, Johan Fabry, Sven Van Caekenberghe: Enterprise Pharo. This is the third volume of a series of books covering Pharo and its community.
http://files.pharo.org/books/enterprise-pharo/

[9] Book: Alexandre Bergel, Agile Visualization:
http://agilevisualization.com

[10] Miguel Campusano and Johan Fabry: "Live Robot Programming: The Language, its Implementation, and Robot API Independence". Science of Computer Programming, Elsevier. Received 8 September 2015, Revised 28 March 2016, Accepted 9 June 2016, Available online 17 June 2016.
http://dx.doi.org/10.1016/j.scico.2016.06.002

[11] Leonardo Humberto Silva, Marco Tulio Valente, Nicolas Anquetil, Daniel Carlos Hovadick Flix, Alexandre Bergel, Anne Etien. "JSClassFinder: A Tool to Detect Class-like Structures in JavaScript". Proceedings of Brazilian Conference on Software: Theory and Practice (CBSOFT'15)
https://hal.inria.fr/hal-01369705

[12] Juan Pablo Sandoval Alcocer, Marcus Denker, Alexandre Bergel, Yasett Acurana. "Dynamically Composing Collection Operations through Collection Promises'. In: Proceedings of IWST16.
https://hal.inria.fr/hal-01358347

[13] Ronie Salgado, Stephane Ducasse "Lowcode: Extending Pharo with C Types to Improve Performance". In: Proceedings of IWST16.
https://hal.inria.fr/hal-01353884

[14] Johan Fabry and Stephen Sinclair: "Interactive Visualizations for Testing Physics Engines in Robotics". Proceedings of 4th IEEE working Conference on Software Visualization (Tool paper) 3-4 Oct 2016. IEEE Computer Society Press

In preparation:

[15] Leonardo Humberto Silva, Marco Tulio Valente, Alexandre Bergel, Nicolas Anquetil, and Anne Etien "Identifying Classes in Legacy JavaScript Source Code' In submission to JSS

[16] Book in preparation about Spec.
https://ci.inria.fr/pharo-contribution/view/Books/job/BuildingUIWithSpec/

5 Future of the partnership

Research can not only be done remotely. Close proximity is the very important, it allows discussions and the gestation of new research ideas and projects.

The work on both Moose, Pharo and many of the technologies worked on during PLOMO2 will continue. Not being able to prolongate the Associated Team will make it harder to finance visits, but we will try to find new sources of funding.