Dear Member of the Office for Research,

This letter contains my review of the PhD thesis titled "Supporting Multiple Languages in Virtual Machines", authored by Jan Vranyi from the Faculty of Information Technologies, Czech Technical University in Prague.

**Main contributions.** The PhD thesis has three main contributions, which expand the scientific knowledge in their respective topic:

- **Debugging architecture:** Debugging facilities are clearly underestimated by the majority of programming environment providers. Even though Smalltalk clearly goes above what the current mainstream environments offer, research space remains wide. Jan's work proposes an architecture that is open, flexible and extensible.

- **Meta-object protocols defined in the virtual machine:** An implementation of an efficient meta-object protocol is provided. This work is a nice demonstration that having a fast MOP is feasible. In addition to nice benchmarks, several method lookups are defined.

- **Exploration of Selector Namespaces:** Selector namespace is an appealing mechanism offered by Modular Smalltalk, S# and Gemstone. However, no open implementation is available and scientifically described. Jan's thesis provides an appealing implementation of selector namespace, which hasn't been done before.

**Questions.** After reading the PhD, I had a number of questions which were left unanswered. Unfortunately, I will not be able to attend Jan's defense (lack of money) I will however appreciate a written answer for each of them.

- If now I would like to implement your MOP in a different language, let's say Pharo. What are the requirements I must fulfill in order to realize the port?
• You have done a great work on providing the essential mechanism for language semantic and execution interoperability. However, you (deliberately) left the interaction with existing library out of your effort. Is there a reason for this beside technical consideration?

• What were the crucial and most difficult points in your validation that you have encountered?

Comments. Regarding the writing style, here are some recommendations:

- Page 3: your abstract should summarize the solution you said you have. Just reading the abstract must give me a complete summary of what you have done.

- Page 49: “Consequently, debugging of multiple-language applications is more resource-costly and error-prone” => this is a strong statement. Whereas I kind of share your feelings, I feel you should temper your statement

- Page 59: “We think that changing the VM is not the way to go” => This has little scientific motivation. You could argue that modifying the virtual machine has a significant cost and is not easily achieved. Especially when comes experimentations on the right semantic you want to give. “This approach also does not scale well.” => Again, a poor justification is provided.

- Page 60: The list of requirements that is supposed to motivate the design of your VM-Level MOP must be refined. You cannot leave it like this. What you describe as “requirements” are actually “constraints”. Your requirements must be related to what your MOP should do.

- Page 65, Section 5.5.4: a short summary about performance would be welcome.

- Page 76: Sections 6.4.1 and 6.4.2 are very brief. I would have liked to read a bit more about the application of Perseus. I remember you told me that Izar has many users. Maybe you could valorize it a bit more by telling who are your users and how many.

- Page 80: “It is said that such a code is very rare (less than 5% of the whole program)” => Do you have any reference for this?

- Page 89: The title “a Decent Ruby Implementation” is a bit provocative. You could rename it into “a fast Ruby implementation” or something related.

- Page 89: “Ruby is a very dynamic scripting language, more dynamic than one may think.” is a very informal sentence. It does not add much to the story.

- Page 98, Section 8.5.1: You mention that one reason that your implementation does not run Rail is the lack of library porting. This strongly suggests that interoperability with foreign library is a corner stone in your story. However, it hasn’t been considered in your thesis. Is there a reason for this?
**Minor comments.** I spotted a number of minor flaws:

- Page 3: “in past few years” => “in the past few years”
- Page 5: “Berger” => “Bergel” :-)
- Page 42: “DebugggerAdapter”
- Page 45: “An Useru interface”
- Page 61: “lookup object lookup algorithm” => maybe the second “lookup” is not necessary
- Page 64: “inbetween” => “in-between”
- Page 76: “an Smalltalk” => “a Smalltalk”, “provides unified” => “provides an unified”
- Page 77: “it provides fast optimized VM” => “it provides a fast optimized VM”
- Page 80: “of the code.” => “of the code.”
- Page 101: “involves wide variety” => “involves a wide variety”

**Conclusion.** I recommend the thesis to be admitted for defense. This Phd thesis presents a significant and deep effort in the field tackled by Jan. Even though, I would have loved to see a stronger scientific validation, Jan explored a difficult and important research track.
A Review of the PhD Thesis of Jan Vraný Entitled
"Supporting Multiple Languages in Virtual Machines"

The thesis presents results of research that Jan Vraný has conducted in the area of interpreted programming languages and their virtual machines. He concentrates on multiple languages support in virtual machines. This area of research is currently quite active. It can be demonstrated by various attempts to implement and use e.g. dynamic languages on top of Java virtual machine. Another example is a multiple languages support by Microsoft's CLR. Using multiple languages in virtual machines is connected with a lot of problems and it still poses a lot of research challenges both of a theoretical as well as practical nature.

The work of Jan Vraný addresses two main problems:

(1) Inability of debuggers to cope with programs written in several languages using several different language runtimes and,

(2) inflexibility of modern virtual machines to handle different language semantics in an efficient way. Jan Vraný proposes solutions of these problems and validates them by implementations in Smalltalk.

Motivations for the work are well described in the context of recent research results in the area of object oriented software composition, such as traits, classboxes, and message selector namespaces, as well as in the area of introducing new languages running on existing virtual machines such as JVM, CLR and Smalltalk/X. Autor quite clearly argues that there is a need for multi-language programming environments. He also shows that the current virtual machine support for hosting multiple languages is limited.

Jan Vraný proposes solutions for the two above mentioned problems. The main contributions of the work are:

(1) Proposal of Debuggable Interpreter Design Pattern and Perseus framework allowing for the construction of debuggable language interpreters.

(2) Proposal of Unified Debugger Architecture capable of debugging the programs written in multiple languages using several different language runtimes.

(3) Proposal of Virtual Machine Level Metaobject Protocol for Customizing Method Lookup which allows programmers to alter the method lookup semantics from outside the virtual machine. It has been designed with respect to VM-level optimizations of the method dispatch and therefore preserves good performance.

Proposed approaches have been implemented and evaluated in using Smalltalk/X and an experimental Ruby implementation for Smalltalk/X VM. Hence, all the main results were sufficiently validated.
The results are original, improving the state of the art in the given area of research. The way Jan Vrany worked on these results is clearly appropriate. The results presented in the thesis were published in a number of reviewed papers at multiple international workshops and conferences. Hence, the results were published in a sufficient way. The thesis is written in English whose quality is good. All of the thesis is well understandable. The original results of the author are presented in an quite precise way. Minor drawbacks are some typos and questionable statements such as “a breakpoint is a signal”.

I have a question that could be discussed within the defense of the thesis:

- The work is clearly important for smalltalkers dealing with novel approaches to composition of software. Can you explain whether and how the work can be important for non-smalltalk (e.g. java or c#) programmers and researchers? Are there some differences in the constructions of VMs for Smalltalk and, say, Java, which make it hard to mimic your Smalltalk solutions in other VMs?

To sum up, I consider the PhD thesis of Jan Vrany to be a work containing a number of valuable and original scientific results. That is why, I recommend the work to be accepted for the defense and upon its successful completion, Jan Vrany to be assigned the PhD degree.

Brno, November 14, 2010

Doc. Ing. Vladimír Janoušek, PhD.
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66, Brno
Report for the Ph.D.:
Supporting Multiple Languages in Virtual Machines
defended by Jan Vrany

The context of this Ph.D. is central to object-oriented programming languages and their implementations. Object-oriented language execution and in particular dynamic object-oriented languages requires complex virtual machine infrastructure to achieve high and optimal execution speed. Now at the same time systems should constantly evolve to adapt to new requirements and the key point of this Ph.D. is that languages follow the same rule. In object-oriented languages, method lookup is then an important point to provide flexibility and virtual machines should be more open and language independent.

The Ph.D. addresses the problem of the introduction of flexible execution entry points in object-oriented virtual machines and the infrastructure to sustain it as well as infrastructure to debug program.

The Ph.D. document has 107 pages. It uses 72 citations. The Ph.D. is its current form is structured around four main parts. The first one sets up the context of the work and is structured in 2 chapters. The second presents the concepts of the approaches: it is structured in three chapters. The third part presents the validation of the Ph.D. and is decomposed in three chapters. The final part is an annex containing benchmarks.

In the motivation part, J. Vrany presents several needs for such extensions: traits, classboxes, selector namespaces. While classbox semantics is inherently dynamic and based on a change of method lookup, the two others, traits and selector namespaces can be implemented without extending method lookup. Traits can statically compiled and selector namespaces can be implemented with a different implementation of Symbol. Now this point does not weaken the problem stressed by the motivation.

The third chapter introduces the debuggable interpreter design pattern which introduces some hooks to plug debuggable services. The general architecture is presented in the following Chapter which presents how the multi-language programs could be debugged using such an architecture. While the hypotheses made were not put up front, they are listed: to work message passing should be controlled and the language in which the message has been sent should be identified. An open question is if the first hypothesis does not imply that languages should be implemented in the same language than the one of the debugger and what is the cost in terms of communication/synchronization between the execution engines if this is not the case.
The chapter five presents an important proposition of the Ph.D.: a meta-object protocol for extending the virtual machine lookup from the executed language itself. It is based on the introduction of an object responsible for the lookup and a lookup point. The important aspect of this work is that it has a low impact on performance. M. Jan Vrany mentions that his protocol allows interception at call and at receiver site but the presentation is not clear.

The following chapter presents the validation of the previous chapter. First with an explanation of some technical aspects of Perseus and second with the application of the protocol in Smalltalk/X. M. Jan Vrany payed attention to offer a solid experimental validation. Now it seems that the benchmarks show that the virtual machine extension is much more costly on real cases such as a browser. Therefore it is not clear that such extension point can be used in general cases. Having a deeper discussion would have been adequate.

The contributions of the Ph.D. are: an unified debugger architecture based on the debuggable interpreter design pattern, the implementation of this architecture into the Perseus framework, a virtual machine level metaobject protocol for method lookup customization and its implementation in a real environment.

M. Jan Vrany is clearly a young and promising researcher, as he demonstrated with his publications and validations. Even if some questions are not clearly answered, I consider that M. Jan Vrany wrote a good Ph.D. and for all these reasons, I recommend the defense of his Ph.D.

Lille, le 10 November 2010.