Institut für Systemsoftware
O.Univ.-Prof. Dr. Hanspeter Mössenböck

JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Compiling Scheme Programs to the Multilanguage Virtual Machine

Master thesis for ...
Matrikelnummer: ...
Email: ...

The Da Vinci Machine Project (also known as Multilanguage Virtual Machine, MLVM) is a research project driven by Sun Microsystems that explores how support for highly dynamic languages can be added to the Java HotSpot™ VM. Some of the enhancements currently under development are *tail calls*, *continuations* and *anonymous classes* (for a more comprehensive list see [1]). In order to prove the viability of the current work, a proof-of-concept implementation of a dynamic, functional language such as *Scheme* [2, 3] is needed.

Goals

Write a compiler that

- compiles Scheme programs to MLVM bytecodes
- makes use of hard tail calls
- makes use of anonymous classes
- implements full continuations
- optionally makes use of other new features (e.g. method handles).

Additionally, the following measurements should be performed:

- How does your implementation perform in comparison to other implementations?
- What improvements (runtime/space/features) are facilitated by the new MLVM features?

The compiler need not implement the full Scheme language—it should however be complete enough to allow it to run reasonably sized real-world applications.

Your work plan should include regular meetings with your supervisor. The implementation of the compiler should be thoroughly tested and documented and should conform to commonly accepted style guidelines .

References

[1] The Da Vinci Machine Project: http://wikis.sun.com/display/mlvm/Home

[2] Scheme: http://en.wikipedia.org/wiki/Scheme_%28programming_language%29

[3] Schemers.org: http://www.schemers.org/

[4] jScheme (Java, small): http://jscheme.sourceforge.net/jscheme/main.html

[5] tinyScheme (native, very small): http://tinyscheme.sourceforge.net/

[6] Bigloo (native, large): http://www-sop.inria.fr/mimosa/fp/Bigloo/

Contact: Prof. Dr. Hanspeter Mössenböck

Start: as soon as possible