Project Proposal

Dynamic Software Update through a Versioning and Compiler Library

Stefano Cherubin

```
Politecnico di Milano

Size of the project:  1 student

Reference person:  Stefano Cherubin <stefano.cherubin@polimi.it>
```

Abstract

```
libVersioningCompiler is a library developed at Politecnico di Milano.  Its purpose is to allow
a program to compile external source code during its execution, dynamically load it and execute
it.
   Goal of the project is to build a prototype of Dynamic Software Update mechanism based on
the LARA domain specific language while exploiting libVersioningCompiler APIs.
```

Pre-requirements

- Basic knowledge of C language

- Intermediate knowledge of C++11 language and idioms

- Basic Linux skills

- [preferred] previous experience with LARA or javascript-like languages

Involved Technologies and Frameworks

- C++11

- Linux

- LARA

Detailed description

libVersioningCompiler is a library developed at Politecnico di Milano whose purpose
is to allow a C++ program to compile external source code during its execution, dynamically
load it and execute it.

This capability is interesting from a software engineering point of view because
it allows to dynamically reconfigure and/or update a piece of software while it is
still running. We assume the programmer writing a software that should be dynamically
updated will specify for each reconfigurable function call which policy has to be applied
in chosing the version that should be executed. A non-exaustive list of policies can
be: most recent version, exact version with given ID, most recent version which has
a given (list of) tag. Updates come as messages through socket.

The goal of the project is to think about these policy, their definition and their
expression, in order to implement a Dynamic Software Update framework compliant with
the above mentioned scenario.

LARA is a Domain Specific Language designed to target HPC systems. It simplifies
optimization operations on C/C++ source code. Write a LARA aspect to automatically
insert the instrumentation code which is needed by the DSU prototype.

At the end of the project a working prototype must be provided.