HSL Interpreter Rewrite

Architectural Design and Prototype of an Interpreter for a Scripting-Language used in Robotic Laboratory Automation

Graduate

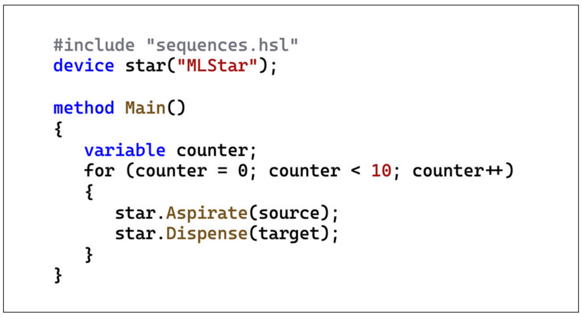
Jann Marco Flepp

Introduction: Hamilton Bonaduz utilizes the VENUS software platform with the HSL (Hamilton Standard Language) interpreter to operate advanced pipetting robots. However, the current interpreter, built long ago in C++ with Lex & Yacc tooling and COM interfaces, struggles with maintainability and extensibility, limiting its adaptability to modern practices and new developments on the VENUS platform.
This thesis aims to resolve these issues by designing and developing a new HSL interpreter.

Approach / Technology: The new HSL interpreter is developed using modern software engineering practices to address the limitations of the existing system. The chosen technology stack includes C# and ANTLR, a powerful parser generator, to ensure a maintainable and extensible codebase.
The thesis adopts a structured development approach that employs the following methods: The requirements analysis is conducted through interviews and document studies of the existing system. The architectural design utilizes C4 modeling to ensure a clear and structured design. The implementation proves the feasibility of the proposed architecture. The quality assurance includes a comprehensive suite of unit, integration, and manual tests to ensure that the developed system meets a high level of quality.

Result: The main outcome of this thesis is a clean software engineering documentation from requirements to architecture and quality assurance of a new interpreter. It demonstrates how the issues of the current system can be resolved with modern design and techniques while maintaining compatibility with the current implementation.
The secondary result is a prototype that successfully demonstrates the viability of the proposed architecture. It can parse and interpret a subset of HSL, and can send simple commands to instruments. It is seamlessly integrated into the VENUS platform and, in addition, features a command-line interface capable of running on non-Windows platforms.

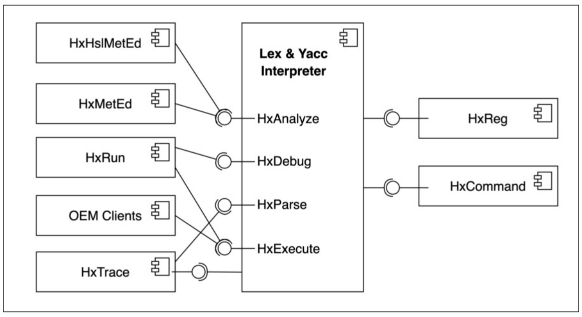
HSL Code Example
Own presentment



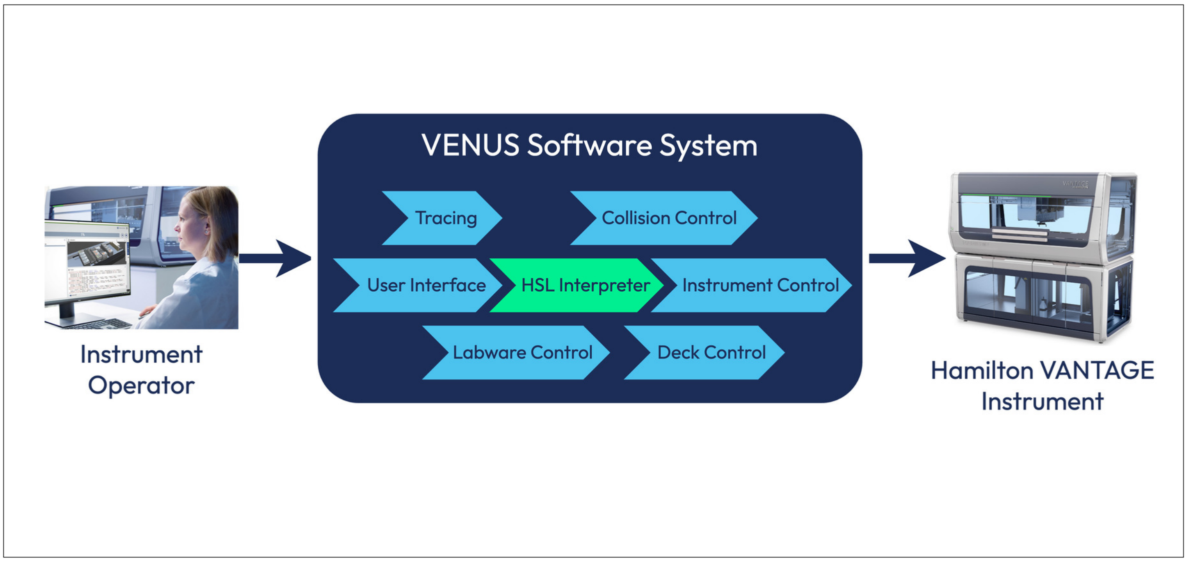
```
#include "sequences.hsl"
device star("MLStar");

method Main()
{
   variable counter;
   for (counter = 0; counter < 10; counter++)
   {
      star.Aspirate(source);
      star.Dispense(target);
   }
}
```

VENUS systems Interacting with the HSL interpreter
Own presentment



Overview of the Hamilton VENUS software platform
Own presentment, Images by Hamilton Bonaduz AG



Advisor
Philipp Kramer

Co-Examiner
Christian Marrocco, Triamec Motion AG, Arni AG, AG

Subject Area
Software

Project Partner
Hamilton Bonaduz AG, Bonaduz, GR