Perl 6 & Parrot Proposal

Introduction

The Perl 6 project stands at a critical juncture. The overall design for Perl 6 is nearly complete. The core implementation, the Parrot virtual machine, is nearing completion. What's missing is a comparatively small component to parse Perl 6 syntax and translate it to a form native to Parrot. This small but critical gap is the difference between a good idea on paper and a usable language. Development will progress no matter what, but hiring a few key developers will accelerate the pace significantly.

This proposal outlines the resources needed to complete Perl 6. The larger project has three target goals: a 1.0 release of Parrot, a development release of Perl 6, and a production release of Perl 6. The development release of Perl 6 represents the point at which other projects can test Perl 6 for their distributions and write software that targets it. More importantly, it will offer a stable platform for module-level developers to build upon, so they can begin porting CPAN.

The 1.0 release of Parrot will have a stable API and promise backward binary compatibility for all future versions of Parrot. Further internals development is anticipated after the release, but these changes will be treated as modifications to a production system with minimal impact on users and graceful deprecation cycles.

Design

The design of Perl 6 isn't set in stone, but the major building blocks are already in place. The remaining work is largely a matter of filling in details.

Because we can't expect Perl 6 design work to stop while implementation proceeds (even Perl 5 continues to evolve), we'll take a snapshot of the current state of the Perl 6 language specification at the start of the project. Implementation efforts for the development release will target the snapshot. This eases the pressure on developers to accommodate shifts in requirements and speeds up the development process. Larry Wall is currently completing a set of high-level summaries of the design (the "Synopses") that will serve as the authority for the snapshot. The production release will integrate any further design changes.

Grammar Engine

The Perl 6 grammar engine (analogous to Perl 5's "regular expression engine") is the heart of the Perl 6 interpreter. A Perl 6 grammar will parse Perl 6 syntax and produce a Parrot abstract syntax tree (AST). The Parrot virtual machine provides the remaining functionality of the interpreter.

The grammar engine will be implemented as a standard Parrot compiler module, so grammars are compiled as an independent language and can be stored as precompiled bytecode. The grammar engine will be accessible from any language running on Parrot.

Development of the grammar engine is divided into 8 milestones. A stable snapshot will be released on the completion of each milestone.

Milestone 1 - complete the engine's parser. To meet this milestone the core of the grammar engine and a Perl 6 grammar will be developed and tested against example Perl 6 code.	US $35,000
Milestone 2 - extend the grammar engine to produce Parrot AST. The core features of the grammar engine will be completed in this milestone.	US $35,000
Milestone 3 - expand the grammar engine to cover the full set of Perl 6 features from the design snapshot. This milestone will include extensive testing to ensure that the Parrot AST output and the compiled Parrot bytecode meet the specification.	US $35,000
Milestone 4 - release the development version of Perl 6. The focus of this milestone is preparing for the release, including bug fixes, extending test coverage, and documentation review. No new features will be added.	US $35,000
Milestone 5 - update the implementation for any design changes. These changes should be relatively small, so any excess funded developer time will be rolled over to milestones 6-8.	US $35,000
Milestone 6 - complete the Perl 5 to Perl 6 translator. This milestone is independent of all other milestones and may proceed in parallel with the rest of the project.	US $35,000
Milestone 7 - implement a Perl 5 regular expression module for Parrot along the same lines as the Perl 6 grammar engine. This will allow the use of current Perl 5 regular expression syntax with the new engine. This milestone depends on the grammar engine milestones 1 and 2, but may be developed by an independent group of volunteers during milestones 3 and 4.	US $35,000
Milestone 8 - release the stable Perl 6.0.0. As in milestone 4, the focus here is preparing for the release. It will concentrate on bug fixes, testing, and documentation, not new development. Several release candidates will be distributed to the general public for testing and review before the final release.	US $35,000

Of these milestones, 1, 6, and 7 are already partially completed.

Parrot

Parrot is the virtual machine (VM) that will run Perl 6. Its register-based design with object oriented internal types is optimized for the needs of dynamic languages such as Perl. This differentiates it from other VMs such as Java's JVM and .NET's CLR which are primarily aimed at static languages. Parrot can run bytecode generated in memory by the Perl 6 compiler, or bytecode read in from disk, which allows Perl 6 code to be distributed in pre-compiled form.

Parrot development is quite advanced at this point, but several key components need to be accelerated. The following subsystems are currently in various stages of development and will need to be completed before the development release of Perl 6:

- IO
- Events
- Threads
- Character set handling
- AST interface
- Object support
- Exceptions
- Multimethod dispatch
- Garbage collection

Parrot development during the first year splits into 4 milestones. Each Parrot milestone provides a level of functionality necessary to complete the parallel milestone for the grammar engine.

Milestone 1 - design the APIs for all critical subsystems. The first grammar engine milestone is relatively independent of Parrot, but the APIs must be defined before work can proceed on the second grammar engine milestone.	US $35,000
Milestone 2 - provide a stub implementation for all critical subsystems. It will provide enough functionality to build and run each system and to support the core features of the Perl 6 grammar.	US $35,000
Milestone 3 - complete full functionality of all critical subsystems.	US $35,000
Milestone 4 - release Parrot 1.0. The work for this milestone focuses on debugging the subsystems, extending documentation, and refining the installation process.	US $35,000

Of the subsystems targeted by this proposal, IO, events, threads, character set handling, exceptions, and garbage collection have already achieved milestone 2. Object support is approaching milestone 3. The AST interface and multimethod dispatch are approaching milestone 1.

Ponie

This proposal does not include the Ponie project (Perl 5 on Parrot). Ponie development is currently funded by donated developer time from Fotango. The accelerated schedule for Parrot development proposed here will also benefit Ponie.

Timeline

No timeline can be guaranteed as long as the project operates solely on volunteer labor. Each milestone represents approximately 6 "developer-months", planned to be completed by two developers working simultaneously. This works out to about US $200 for one "developer-day".

Project Staff

The project will employ four people full-time: two for Parrot and two for the grammar engine and Perl 6.

Leopold Tötsch is the current pumpking (lead developer) for Parrot. He is an independent software developer living in Austria. He first started working with computers in 1976 and has explored and developed open source software since 1991. Leo will be the primary Parrot developer.

Chip Salzenberg is the chief architect of Parrot. He was the pumpking (lead developer) for Perl 5.003_08 through 5.004 and 5.004_05 in 1996-1999. In 1998 he started project Topaz, a reimplementation of Perl 5 in C++. The strengths and weaknesses of Topaz had significant impact on the design and implementation of Parrot. Chip will be the secondary Parrot developer, focusing on stability and portability.

Patrick Michaud is the pumpking for Perl 6. He has been a Professor of Computer Science at Texas A&M University - Corpus Christi since 1990 and was the program director of the Texas Coastal Ocean Observation Network (TCOON) from 1996 until 2003. He is currently on a leave of absence from the University to spend time on open source development. Patrick will be the primary developer of the grammar engine.

Larry Wall is the original creator of Perl and the language designer for Perl 6. In addition to continuing design work, he will assure that the implementation meets the requirements of the design. He will also be the primary developer of the Perl 5 to Perl 6 translator.