Designing Lisp for Ubiquity

Making Lisp programming mainstream again

Why do I want to use Lisp?

- Code is data
- Cool symbolic runtime library
- Listener window

Attributes of Lisp Systems

- Weakly typed
- Rich OO runtime (CLOS)
- Code is data
- Reflection
- Dynamic evaluation
- Meta programming
- IDE with data/control inspectors
- Rich library / Frameworks (CLIM…)

Other Languages Squeeze Lisp

- Untyped scripting languages
  - Python
  - Ruby
  - JavaScript/PHP
- Strongly typed dynamic languages
  - C#, VB.Net…
  - Java
- Data language: XML

Lisp is Unique

- It has all of the good attributes
- Very few performance compromise
- This is a strength and a weakness
  - Tends to be self sufficient
  - Does not play nice with native languages

Lisp Interoperability

- Treats native languages as second class
- Suffers from representation issues (full machine word integer, floating point…)
- Native -> Lisp interop is generally hoky
- Does not integrate with popular frameworks well
- Native tools (debuggers/linkers) don’t support Lisp

2 significant platforms

- New dynamic language platforms
  - Languages and IDE/Toolset
- Java
  - Scripting languages
- .NET
  - Cobol, Eiffel
  - Scripting languages (IronPython)

Lisp as a Native Language

- Use native primitive formats (int, float…)
- Use native object representation
- Use same byte code / binary files
- Good integration with dominant IDE toolset

What are the advantages?

- A component could be written in Lisp and none of their “users” notice
  - Can use standard IDE/Tools
  - Can use/extend existing Frameworks
- Language choice can be fine grained
  - Use Lisp for its unique strengths
- Don’t fight irrelevant battles (FFT benchmark)

What is the cost?

- Weakly typing is much slower
- OO concepts lacking

My opinion on a native Lisp

- No weak typing, explicit declarations.
- No OO extensions (subset of CLOS)
- Keep MOP.
- Keep Closures.
- Use runtime code generation for dynamic features.
- Keep subset of Common Lisp runtime (strings aren’t mutable…)
- Integrated with native IDE

Conclusion

- Lisp has a role to play on new platforms
  - XML DOM navigation
- If a decent Lisp does not appear, the void will be filled up sooner or later by other languages/libraries