Exploring Advanced Medical Image Processing & Related Informatics Using Ruby + UR/Web + Clifford Algebra + QRNG + Machine Learning + Wavelets.

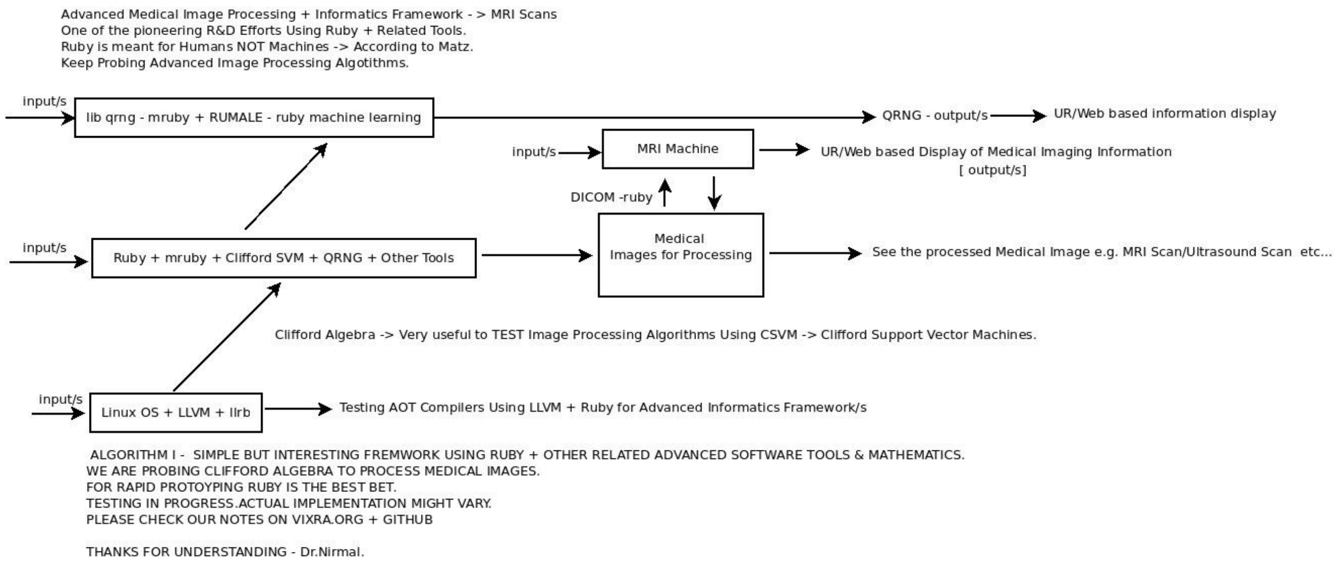
[Advanced Information Processing Using UR/Web – One of the Pioneering R&D Efforts]

Nirmal – Informatics R&D – USA/UK/Israel/Japan/BRICS Group of Nations.
Current Member – ante Inst UTD Dallas TX USA.
Contact_info – hmfg2014@gmail.com

[I] Main Idea + Inspiration + Introduction :

Ruby/mRuby w.r.t Tomographic Image Processing R&D Algorithms Using Wavelets + Support Vector Machines [SVMs] + QRNG [lib-qrng-mruby] + Clifford Algebra [CA] → Towards Testing Medical Instrument/s + Smart Devices + IoT + HPC Heterogeneous Systems & Computation.

[II] Ruby Based R&D Medical Image Processing + Informatics Framework :


Advanced Medical Image Processing + Informatics Framework - > MRI Scans
One of the pioneering R&D Efforts Using Ruby + Related Tools.
Ruby is meant for Humans NOT Machines -> According to Matz.
Keep Probing Advanced Image Processing Algotithms.
input/s
lib qrng - mruby + RUMALE - ruby machine learning
QRNG - output/s
UR/Web based information display
input/s
MRI Machine
UR/Web based Display of Medical Imaging Information [output/s]
DICOM -ruby
input/s
Ruby + mruby + Clifford SVM + QRNG + Other Tools
Medical Images for Processing
See the processed Medical Image e.g. MRI Scan/Ultrasound Scan etc...
Clifford Algebra -> Very useful to TEST Image Processing Algorithms Using CSVM -> Clifford Support Vector Machines.
input/s
Linux OS + LLVM + llrb
Testing AOT Compilers Using LLVM + Ruby for Advanced Informatics Framework/s
ALGORITHM I - SIMPLE BUT INTERESTING FREMWORK USING RUBY + OTHER RELATED ADVANCED SOFTWARE TOOLS & MATHEMATICS.
WE ARE PROBING CLIFFORD ALGEBRA TO PROCESS MEDICAL IMAGES.
FOR RAPID PROTOYPING RUBY IS THE BEST BET.
TESTING IN PROGRESS.ACTUAL IMPLEMENTATION MIGHT VARY.
PLEASE CHECK OUR NOTES ON VIXRA.ORG + GITHUB
THANKS FOR UNDERSTANDING - Dr.Nirmal.

[Figure I – Simple Algorithm I]

“**Ur** is a programming language in the tradition of ML and Haskell, but featuring a significantly richer type system. Ur is functional, pure, statically typed, and strict.

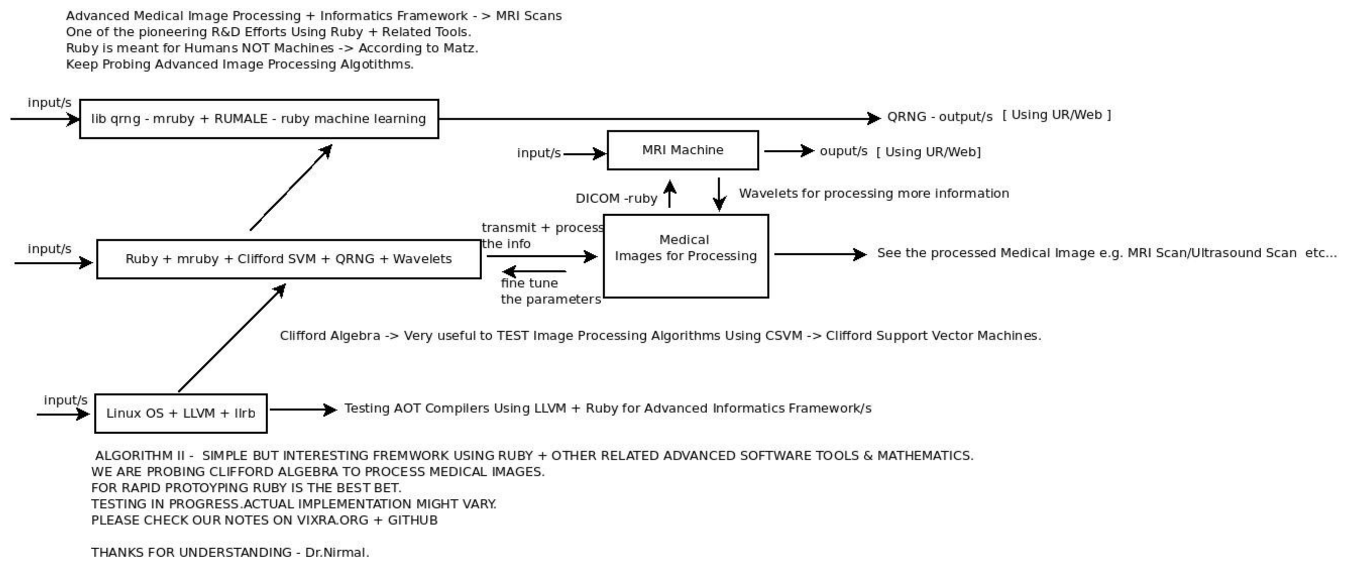
Ur supports a powerful kind of **metaprogramming** based on **row types**.**Ur/Web** is Ur plus a special standard library and associated rules for parsing and optimization. Ur/Web supports construction of dynamic web applications backed by SQL databases.

The signature of the standard library is such that well-typed Ur/Web programs "don't go wrong" in a very broad sense. Not only do they not crash during particular page generations, but they also may not:

- Suffer from any kinds of code-injection attacks
- Return invalid HTML
- Contain dead intra-application links
- Have mismatches between HTML forms and the fields expected by their handlers
- Include client-side code that makes incorrect assumptions about the "AJAX"-style services that the remote web server provides
- Attempt invalid SQL queries
- Use improper marshaling or unmarshaling in communication with SQL databases or between browsers and web servers

This type safety is just the foundation of the Ur/Web methodology. It is also possible to use metaprogramming to build significant application pieces by analysis of type structure. For instance, the demo includes an ML-style functor for building an admin interface for an arbitrary SQL table. The type system guarantees that the admin interface sub-application that comes out will always be free of the above-listed bugs, no matter which well-typed table description is given as input.

The Ur/Web compiler also produces very efficient object code that does not use garbage collection. These compiled programs will often be even more efficient than what most programmers would bother to write in C. For example, the standalone web server generated for the demo uses less RAM than the bash shell. The compiler also generates JavaScript versions of client-side code, with no need to write those parts of applications in a different language. The implementation of all this is open source.” **ref [b]**.


input/s
lib qrng - mruby + RUMALE - ruby machine learning
QRNG - output/s [Using UR/Web]
input/s
MRI Machine
ouput/s [Using UR/Web]
DICOM -ruby
Wavelets for processing more information
input/s
Ruby + mruby + Clifford SVM + QRNG + Wavelets
transmit + process the info
fine tune the parameters
Medical Images for Processing
See the processed Medical Image e.g. MRI Scan/Ultrasound Scan etc...
input/s
Linux OS + LLVM + llrb
Testing AOT Compilers Using LLVM + Ruby for Advanced Informatics Framework/s

Advanced Medical Image Processing + Informatics Framework - > MRI Scans
One of the pioneering R&D Efforts Using Ruby + Related Tools.
Ruby is meant for Humans NOT Machines -> According to Matz.
Keep Probing Advanced Image Processing Algotithms.

Clifford Algebra -> Very useful to TEST Image Processing Algorithms Using CSVM -> Clifford Support Vector Machines.

ALGORITHM II - SIMPLE BUT INTERESTING FREMWORK USING RUBY + OTHER RELATED ADVANCED SOFTWARE TOOLS & MATHEMATICS.
WE ARE PROBING CLIFFORD ALGEBRA TO PROCESS MEDICAL IMAGES.
FOR RAPID PROTOYPING RUBY IS THE BEST BET.
TESTING IN PROGRESS.ACTUAL IMPLEMENTATION MIGHT VARY.
PLEASE CHECK OUR NOTES ON VIXRA.ORG + GITHUB

THANKS FOR UNDERSTANDING - Dr.Nirmal.

[Figure II – Simple Algorithm II]

[III] Useful + Important References :

[a] https://github.com/tejdnk-2019-ShortNotes

[b] http://www.impredicative.com/ur/

[c] http://www.impredicative.com/wiki/index.php/Libraries_and_FFI_bindings#uw-ruby

[d] https://frigoeu.github.io/urweb1.html

[IV] Acknowledgment/s : Non-Profit R&D. Inspire Others Always. Sincere Thanks to all WHO made this happen in my LIFE.

[THE END]