

NEUROBRIDGE: AN AI DEVELOPMENT ENVIRONMENT FOR NEUROLOGY

Jeffrey L. Sponsler, MD, MS
Alaska Brain Center, LLC
4551 E Bogard Rd Wasilla AK 99654
jsponsler@akbraincenter.com

ABSTRACT

The development of AI tools for expert systems usually begins with creation of a suite of tools. A neurology-specific AI toolkit is not found in the literature. A software development tool kit, the NEUROBRIDGE, has been developed using Common Lisp. The tool suite includes object-oriented anatomic atlas, natural language processing tools, a data extraction interface between parse trees and database, a logic programming system, an interface between electronic medical records and the database, graphical presentations, and test benchmarks. The workbench provides an infrastructure for neurological AI systems. NEUROBRIDGE was necessary and essential to development of neurological expert systems and should provide a robust framework for future systems evolution.

KEY WORDS

Expert System, Artificial Intelligence, Lisp, Prolog, Neurology

1. Introduction

The automation of data analysis is required in the current era to increase efficiency and accuracy of clinical systems to improve patient diagnosis and treatment.

Many AI environments have been developed in the past. Some examples include POPLOG [1], Babylon [2], and KEDE [3]. These systems include semantic nets, natural language processing, and rules but do not apply to neuroscience. Literature search reveals no AI development environments that have a neuroscience focus.

Development specification: The goals of NEUROBRIDGE development included creation of a logic programming language in Lisp, a framework for definition of neuro-anatomic structures, a natural language parsing system, a rule based expert system shell (using logic programming), and an electronic medical record specific to neurology. The secondary goal of NEUROBRIDGE included development of the knowledge definitions (brain, nerves, muscles, arteries, pathways) and expert system rules to solve specific

medical problems (including seizure classification, stroke diagnosis, brachial plexus injury diagnosis). Secondary goals also included grammar rules and grammar selection rules for parsing sentences. Rapid prototyping development was used for this project.

To address the specifications above, the NEUROBRIDGE has been created. This toolkit is described in this report. The components of the NEUROBRIDGE are described below. The components include Prolisp [4], disease benchmarks, HPARSER [5], parse tree data extraction engine [5], digital brain atlas, StrokeDx expert system shell [4, 6], and SYNAPS diagnostic engine [7]. Knowledge components were developed using NEUROBRIDGE and are described below.

2. Resources

Programming Languages

NEUROBRIDGE was coded in Common Lisp [8], Common Lisp Object System (CLOS) [9], and Prolisp [4]. Prolisp incorporates concepts from Prolog [10, 11].

Knowledge Bases and Databases

The NEUROBRIDGE supports these knowledge systems: the Neuro-Anatomic Atlas [12], grammar augmented transition networks, parser constraint knowledge base, parser grammar selection rules, patient examination database, an object-oriented knowledge base for electromyography data analysis and diagnosis, and a stroke diagnosis expert system (StrokeDx) rule base. The Unified Medical Language System (UMLS) from the National Library of Medicine [13] has been incorporated as lexicon for the HPARSER system. Each is described below.

3. Neuro-Anatomic Atlas

An object-oriented knowledge base of components of the central nervous system (CNS) and the peripheral nervous system (PNS) has been created. The Neuro-Anatomic

Atlas (NAA) uses Lisp and CLOS to instantiate a network of objects. The NAA includes knowledge of the corticospinal system, dorsal column medial lemniscus (DCML) system, the brachial plexus, the basal ganglia, brainstem and brainstem nuclei, spinal cord, dorsal roots, ventral roots, spinal ganglia, cortical gyri, Brodmann's areas, thalamus, thalamic nuclei, thalamic connections, the trigeminal sensory system, all muscles, all peripheral nerves, and all arteries to the central nervous system. The knowledge base can be accessed by Lisp functions to support clinical decision support that requires detailed anatomic details of the CNS and PNS. A subsystem of the NAA is a knowledge base describing the brachial plexus called PLEXBASE [14]. The NAA network contains over 1500 objects.

An example function call for vessels for the internal capsule follows.

```
(vessels-of 'posterior-limb-internal-capsule-left) →
(<ARTERY lateral striate artery left> <ARTERY
middle cerebral artery left> <ARTERY internal
carotid artery left> <ARTERY common carotid
artery left> <ARTERY aorta>)
```

Another example of a blood vessel query for Brodmann area 4 is reported here:

```
(vessels-of 'area-4-right) →
(<ARTERY rolandic branch right> <ARTERY middle
cerebral artery right> <ARTERY internal carotid
artery right> <ARTERY common carotid artery
right> <ARTERY brachiocephalic trunk right>
<ARTERY aorta>)
```

Definitions for nervous system pathways are created explicitly and are evaluated on system start. Evaluation of definitions creates a table of objects and a memory resident object network. Anatomic names are unique and are dereferenced, in an initialization step, so that pointers are substituted into the objects. An example definition that describes components of the sensory path from the C5 dermatome to the corresponding parietal lobe area follows:

```
(define-sensory-subpathway
 :id 'c5-touch-left
 :component-of 'dorsal-column-medial-lemniscus-
system-left
 :modality 'touch
 :segments '(dermatome-c5-touch-left drg-c5-touch-
left nucleus-cuneatus-c5-touch-left vpl-c5-touch-
right sensory-area-1-c5-right))
```

Graphics: Graphical user interface code was developed for the NAA to present arterial networks, 3D brain images of stroke location, and natural language sentence analysis.

4. Confidence Factors

For numerical representation of truth this system uses the *confidence factor* (CF). The standard convention for a CF is zero represents false, 0.5 represents unknown, and 1.0 represents true. A mathematical operator, *alpha*, is employed in this system [15]. Applied to confidence factors, alpha combines values synergistically. Operator *average* is also used to combine CF's.

5. Logic programming tool PROLISP.

A custom built *Prolog in Lisp* system called **Prolisp** was developed [4]. Prolisp uses CLOS objects to store rules and facts. Prolisp uses *facts*, *rules*, *unification*, and *resolution* in a manner similar to Prolog [8, 9]. Prolisp variables have a question mark prefix (e.g., ?var). The function **define-fact** defines facts where fact is a pattern such as (planet earth). Function **define-rule predicate clauses** defines rules with *predicate* is a pattern and *clauses* are the sub-goals. Prolisp includes an operator “=” that forces unification of a variable and another form. For example, (= ?x 100) will unify ?x with value 100. Prolisp facts have the form (*functor args*) where *functor* is the predicate name and *args* is zero or more arguments. Prolisp rules are defined by this prototype: (define-rule *head body*) where *head* is a predicate pattern and *body* is a list of clauses each of which must be matched/proved for the predicate to succeed. The *and* operator is implicit for each clause group. The *or* operator is implemented as either a second rule with the same head or by use of *define-rules* which supports multiple and-clause groups. The integration of a logic programming module into the Lisp environment provides depth-first deductive retrieval and pattern matching and has facilitated expert systems development. The Prolisp function (**proof pattern**) runs the Prolisp search and takes argument *pattern*. Pattern is the predicate to be proved by search.

```
(define-fact '(planet earth))
(proof '(planet ?x))
?X = EARTH
```

This example defines a rule with only one clause:

```
(define-rule '(sun-has-planets ?x) '((planet ?x)))
(proof '(sun-has-planets ?x))
?X=EARTH
```

Prolisp also incorporates programming concepts from a system called SNARK [16]. These concepts include

rewrite rules and *satisfier* rules. Rewrite rules specify lisp functions that perform elementary operations such as math and return the values to Prolisp. Some rewrite rules will return a list as a pointer which is allowed since Prolisp is Lisp resident. A satisfier rule specifies a lisp function that is evaluated (during run-time) and returns a list of facts for rule processing and these facts are derived from data stored outside of the Prolisp fact base.

Example function *define-rewrite-rule* creates a rewrite-rule wherein a functor alpha-rule maps to Lisp function alpha. A call to the lisp function with values of ?x and ?y returns a value that is assigned to the variable ?z.

```
(define-rewrite-rule '(alpha ?x ?y ?z)
  '((= ?z (alpha-rule ?x ?y)))
  :rewrite-name 'alpha-rule :lisp-function 'alpha
  :arity 2 :name 'ALPHA2)
(proof '(alpha 0.7 0.7 ?x))
?X = 0.84
:PROOF
```

Note that alpha converts two moderately true values to a relatively higher truth value.

A system to diagnosis epilepsy syndromes was coded using Prolisp [17].

Example Prolisp rules: Horner Syndrome and Wallenberg Syndrome

The rules for diagnosis of Horner Syndrome and Wallenberg Syndrome are stated here. Wallenberg syndrome requires Horner Syndrome. Ptosis is eyelid droop, meiosis is abnormally small pupil, and anhidrosis is warm/dry face.

```
(pro:define-rule '(horner-syndrome ?side ?cf) ;; the
predicate
'((has-ptosis ?side ?p-cf) (has-meiosis ?side ?m-cf)
(has-anhidrosis face :right ?a-cf) ;; clauses
```

The top level rule for diagnosis Wallenberg is stated here in an abridged version. Note that predicate horner-syndrome is part of the and-clause group.

```
(pro:define-rule
'(WALLENBERG ?side ?wallenberg-cf)
'((contralateral ?side ?cl-side)
(horner-syndrome ?side ?horner-cf)
(has-limb-ataxia ?anatomy ?side ?ataxia-cf)
(loss-of-pain-sensation face ?side ?pain-cf)
(loss-of-temperature-sensation face ?side ?temp-cf)
...))
```

The function **define-rule-and-default** *predicate and-clauses default-clauses* has been developed. A Prolisp proof will fail if appropriate facts or rules are not found. A proof attempting to diagnose medical conditions must run to completion; proof failure has little diagnostic value. In the absence of patient facts, many StrokeDx rules have associated default rules that allow a proof to complete. The default clauses will succeed and set default confidence (usually 0.5). The *define-rule-and-default* function instantiates a rule, its predicate, its *and-clauses*, and a set of AND-logic default clauses. The predicate is the principle match pattern. And-clauses are evaluated sequentially. If the and-clauses fail, then the default-clauses are processed. The goal is rule success. If a rule fails, the default clauses can rescue the rule, bind default values and default confidence factor, and allows the diagnostic search algorithm to continue. An example follows:

6. Electromyography Analysis

A knowledge based program Electromyography and Nerve Conduction Analysis System (ENCAS) was developed using Lisp and CLOS [18]. This system analyzes test data from EMG and NCS testing and reduced data analysis time from 30 minutes to less than one second. ENCAS accesses the NAA system for knowledge about nerves, muscles, and spinal nerve roots. EMG/NCS knowledge is encoded in the object definitions. EMG knowledge is stored in defined test objects and methods. This system is in active use in the author's clinic.

7. Electronic Medical Record for Neurology

A database management system for neurology clinic patient data was developed and has been documented [19]. This system is in active use in the author's clinic. A report generator can write patient data into a Lisp file format. This file can be loaded into the NEUROBRIDGE database. The file contains functions that when evaluated store the data into the master examination record. Future work will utilize this report generator to test diagnostic systems.

8. StrokeDX

The expert system shell StrokeDx [4] was coded using Prolisp rules and Lisp access functions. The iconic medical diagnostic system MYCIN identified bacterial infections [20]. To exercise rules, case benchmark files were developed. StrokeDx is described below.

Benchmark Cases

A *benchmark* case is a set of patient findings that are characteristic of a specific stroke syndrome. The benchmark cases include these diagnoses: frontal stroke, occipital stroke, Wallenberg syndrome, radial neuropathy, Weber syndrome, Miller-Gubler syndrome, and Brown-Sequard syndrome. Descriptions of these neurological diagnoses can be found in Brazis [21].

CLOS classes include *examination* and *pxdata*. The examination object is a comprehensive storage object. A *pxdata* is a single piece of clinical data (“weakness of right biceps”) and is an n-tuple (*anatomic-structure side quality grade CF*). An example is (triceps left weakness 4 0.7). A Prolisp fact is a pattern for patient data representation. Methods convert patient examination/*pxdata* information to Prolisp facts. The facts have the same n-tuple organization.

Benchmark data includes patient background, examination findings, magnetic resonance imaging (MRI) findings, computed tomography (CT) findings, magnetic resonance angiogram (MRA) findings, computed tomography angiogram (CTA) studies, and laboratory findings. The benchmark cases encode patient demographics, stroke risk factors, examination data (e.g., weak right biceps), MRI image analysis (diffusion restriction left frontal), CT image analysis (hypodensity left frontal), CTA results (occlusion of left middle cerebral artery), and MRA results (occlusion of left middle cerebral artery).

The benchmark signs include sensory data, motor data, reflex data, cranial data, etc. Sensory data (touch, vibration, proprioception, pain, temperature) are restricted to a description with this pattern: modality, side, dermatome, grade, CF. An example: *vibration sensation in the left C5 dermatome is absent and confidence is 1*. At present, this dermatome based system is used because the somatotopic map is preserved from body surface to spinal cord to thalamus to parietal cortex [17].

Motor data are restricted to a description with this pattern: side, muscle, grade, CF. An example: *right triceps muscle has grade zero strength and confidence is 1*. If a rule requires sub-goal of right arm weakness then this will be defined by sub-goals that include some or all of the muscles of that arm.

Benchmark to Prolisp Facts

The initial step is to execute code that takes the benchmark dataset (an object) and for each attribute

create a Prolisp fact. A subset of the fact base (typical count is 200 - 300) is listed below. As with Prolog, facts are patterns on which the theorem prover applies rules in a depth-first search. The facts below include propositions about patient symptoms, signs, and test results. For example the fact (CTA POSTERIOR-CEREBRAL-ARTERY :RIGHT OCCLUDED 1.0) means that the a CT angiogram was done and the right posterior cerebral artery is occluded and this test result has confidence 1.

```
(TOUCH C5-DERMATOME :RIGHT ABSENT 1.0)
(STRENGTH TRICEPS :LEFT 4 1.0)
(CTA POSTERIOR-CEREBRAL-ARTERY :LEFT OPEN 1.0)
(CTA POSTERIOR-CEREBRAL-ARTERY :RIGHT OCCLUDED 1.0)
```

If a fact has confidence zero, the fact is false or the confidence in the fact is extremely low. In general, the StrokeDx software employs the first strategy.

9. Electronic Medical Record Interface

A clinical EMR [19] was coded using PHP [22], MYSQL [23], and APACHE [24]. The EMR includes software to generate data transfer files. The transfer files contain Lisp functions that when evaluated populate data structures in the NEUROBRIDGE. The data transfer involves clinical data being instantiated in the *examination* data structure. The *examination* structure is common to this interface and to the benchmark files. This interface will, in future efforts, support testing of NEUROBRIDGE systems on actual clinical data. Example interface functions are below.

```
(MEDICATION :PATIENT-ID "2930"
:VISITNUMBER "7" :VISITDATE "2011-06-02"
:RXNAME "Synthroid" :DOSE "1.25 MG" :UNITS
"MG" :FREQ "DAILY" :PRN "")
```

10. Rule Set Definitions

A rule set manager has been developed to support Prolisp rule set libraries. The manager uses define-rule-file and define-rule-set. Programming involves defining rule files wherein there are rule and fact definitions. A defined rule-set groups together several rule files. Loading a rule set forces the component rule files to be loaded to build the knowledge base for the diagnostic engine StrokeDx. Future development will allow switching from one rule base to another without reloading.

```
(pro:define-rule-file 'RULES-STROKE
:filename "rules-stroke")
```

```
(pro:define-rule-set 'STROKE-RULE-SET
  :rule-files '(rules-basic rules-stroke rules-cadasil))
```

11. Natural Language Parser for Stroke Cases

In a previous research effort [7], we described the System for Neurological Analysis of Patient Symptoms (SYNAPS), a diagnostic engine that used a network of CNS objects (NAA). The front-end for SYNAPS, HPARSER [5], parsed free text reports and the resulting parse trees were “mined” for patient data. Data mining was via knowledge-based *parse-tree associated constraints*. A constraint was an n-tuple of (grammar-rule algorithm). N-tuple semantics: if a parse tree contains this grammar-rule then run the associated algorithm. In the algorithm, the constraint algorithm analyzes the parse tree and extracts and stores data. For example, if a blood pressure parse tree is discovered, the attached constraint will find the systolic pressure number and store it. Patient data was then used by StrokeDx to compute stroke localizations.

Grammar Set Selection

In the current effort, HPARSER was re-engineered to support *grammar sets* and keyword *grammar set selection*. A grammar set is a relatively small named grammar that can be selected and applied to a sentence. Grammar set selection algorithm is straightforward: pre-scan the sentence for keywords and based on the scan, select a grammar set. For example, if a sentence contains the word biceps, then the biceps grammar set is indexed and applied for parsing. The outcome is that a concise grammar set reduces the search space for a given utterance, speeds search, and improves parse tree discovery. Parse time was reduced (dramatically) and parsing is completed on a given sentence in milliseconds.

The example context definition below specifies that if the words “mri” or “stroke” or “frontal” are detected in the sentence, then access grammar *radiology-grammar* and apply the grammar rule *mri-stroke-right-frontal* to the sentence. This reduces the parser search space and improves time efficiency for the parse. In the event that the parse fails with the restricted grammar, the activated context can be used to provide some data to the other systems. The heuristic: If the key words are found then within some range of error, a data association that might be clinically useful can be obtained. In this example, the presence of the list of *words* may imply that there is “a stroke seen in the right frontal lobe by MRI imaging.”

This matching algorithm need not conform to precise grammar; key word identification alone triggers the heuristic. The goal of maximizing clinical data collection is supported by this heuristic. Since the data is less robust, a confidence factor of 0.75 can be assigned.

```
(define-parser-context      'MRI-STROKE-RIGHT-
FRONTAL
  :atn-id 'radiology-grammar
  :subnet-id 'mri-stroke-right-frontal
  :words '(mri magnetic resonance right stroke
diffusion restriction head frontal))
```

Database Storage Constraint links parser to database.

Definitions of storage constraints link parser grammar rule identifiers to the clinical database. In the example below, the constraint is linked to parser net *mri-stroke-right-frontal*. When this parser net is successful (and is integrated into the parse tree), then data (“right” and “frontal-lobe”) are stored into an object of class *mri-stroke-finding* and added to the *neuro-exam* object.

```
(define-db-storage-constraint
  :id 'MRI-STROKE-RIGHT-FRONTAL
  :atn-id 'radiology-grammar
  :net 'mri-stroke-right-frontal
  :table 'neuro-exam
  :attribute 'mri-stroke-right-frontal
  :storage-class 'mri-stroke-finding
  :store '(:store :right :slot side) (:store frontal-lobe
:slot locus))
```

HPARSER/StrokeDX Results

Patient free text reports were created for frontal stroke, occipital stroke, Wallenberg stroke, and radial neuropathy. A report contains the sentence such as “The right biceps strength was grade zero.” Using HPARSER, each free text case was parsed and via *parse-tree associated constraints* medical findings were obtained and then stored in the format for StrokeDx. New parsing rules were written to include key phrases found such as “MRI shows restricted diffusion in left frontal lobe.” StrokeDx results are documented elsewhere; the system had good diagnostic confidence for these cases. Confidence factors were in general lower than in the other trials; this reflects lower general data collection rate (natural language parsing, a weak method, is more prone to incorrect or missing data).

12. Video EEG Expert System

A system to classify epilepsy syndromes [17] was encoded using NEUROBRIDGE and Prolisp. This system was applied to multiple real cases and was accurate. The system was able to determine brain localization of seizure onset and performed at the level of a trained neurologist.

13. StrokeDx: Diagnosis System

StrokeDx [4, 26] is a diagnostic system with high sensitivity (with complete data sets). StrokeDx uses a robust knowledge framework that is easily extended with new rules. Patient data that is incomplete will trigger defaults (CF 0.5) and diagnostic confidence is less robust. Complete patient data improves diagnostic certainty. The encoding of default rules for missing data has both advantages and disadvantages. Default rules ensure that missing data will not cause search to fail. The default value of 0.5 (unknown) semantically is reasonable because the number does not support or deny a diagnosis. StrokeDx supports rules to diagnosis frontal stroke, occipital stroke, thalamic stroke, Weber syndrome, Millard-Gubler syndrome, Wallenberg syndrome, Brown-Sequard syndrome, radial palsy, and superior trunk brachial plexus lesion.

14. PLEXBASE

As described in another report [27], an object-oriented model of the human brachial plexus called PLEXBASE has been developed. The model includes motor connections, sensory connections, spinal roots, trunks, divisions, cords, nerve branches, muscles, and dermatomes. Functions to access the model were developed. Prolisp rules to reason on the components of the model were written. An example query follows:

```
(proof '(muscle-and-dermatome-share-trunk TERES-MAJOR-RIGHT ?dermatome ?trunk ?hx))
?DERMATOME = DERMATOME-C6-RIGHT
?TRUNK = SUPERIOR-TRUNK-RIGHT
?HX = (MUSCLE-AND-DERMATOME-SHARE-TRUNK SUPERIOR-TRUNK-RIGHT (MOTOR-CONNECTION VENTRAL-HORN-C5-RIGHT SUPERIOR-TRUNK-RIGHT POSTERIOR-DIVISION-SUPERIOR-RIGHT POSTERIOR-CORD-RIGHT LOWER-SUBSCAPULAR-NERVE-RIGHT TERES-MAJOR-RIGHT) (SENSORY-CONNECTION SPINAL-GANGLION-C6-RIGHT SUPERIOR-TRUNK-RIGHT POSTERIOR-DIVISION-SUPERIOR-RIGHT POSTERIOR-CORD-
```

```
RIGHT AXILLARY-NERVE-RIGHT DERMATOME-C6-RIGHT))
```

This query finds a dermatome that shares the same trunk with teres major muscle. The shared trunk is bound to the ?trunk variable and a history structure is also made available. The history includes underlying motor and sensory connections. PLEXBASE rules are able to answer many detailed questions about the brachial plexus and to diagnose lesions of parts of the plexus.

System Performance Evaluation

No independent evaluation of these systems has been performed. The developer is a neuroscientist and is able to assess correctness of knowledge and behavior; this is not sufficient and formal validation is important and is planned. Software speed is good; all programs yield results in less than one second on conventional architecture with the exception of the language parser which can take up to one minute of processor time for non-trivial sentences.

Future Work

The manual coding of benchmark files is tedious. Long range goal is to improve the electronic medical record interface to the benchmark data structures. Once this development has been done, the manual coding may end. In its place would be EMR resident test cases that can be populated with data via EMR user interface. This population would include text report history, symptoms checklist data, laboratory data, medicine information, examination data, and radiology data. Much of this work has been initiated but is incomplete.

For each diagnostic benchmark dataset, a corresponding free text benchmark file is planned. Options for this work include (a) manually encoding the history files or (b) automatically generating history files from the benchmark data structures. This set of natural language benchmarks would then provide test data for the natural language parsing front-end to the StrokeDx system.

President Barack Obama has recently announced BRAIN project [28] to map the human brain and understand its workings. The NEUROBRIDGE contains knowledge of many brain pathways and will contribute to this research effort.

15. Conclusions

A software toolkit called NEUROBRIDGE has been developed to support clinical neuroscience AI systems. These systems provide the infrastructure on which to encode complex anatomic structures of the CNS and PNS,

to develop grammar sets for parsing medical reports, to develop knowledge based diagnostic systems for stroke, seizures, neuromuscular disorders, and neuro-immunologic disorders. NEUROBRIDGE was developed synchronously with the knowledge based systems but could be applicable to any domain since knowledge data is explicit (separate from algorithmic code). The NEUROBRIDGE supports explicit knowledge encoding (rules, networks, objects) and thus facilitates rapid development of new programs. NEUROBRIDGE systems can be deployed as clinical applications and for teaching medical providers. Internet access to NEUROBRIDGE knowledge systems is planned. Applications built on NEUROBRIDGE include stroke expert system, seizure classification expert system, brachial plexus expert system, patient report parser, and an electromyography expert system.

References

- [1] Partridge D, 1991. POPLOG in *Artificial Intelligence and Software Engineering* Ch. 9 p. 203-233. Ablex Pub. Co. NJ.
- [2] Christaller, Di Primio, Schnepf, Voss, 1992. *Babylon: An Open and Portable Development Environment for Expert Systems* Academic Pr 2nd edition, Sep 1992.
- [3] Z Zheng, Wei Li, 1992. Hybrid Knowledge Engineering Development Environment *Int. J. Artif. Intell. Tools* 01, 463 (1992).
- [4] J Sponsler, MD. StrokeDX: A logic programming system to diagnose stroke. *Proceedings of the IASTED Conference for Telehealth*, Innsbruck Austria, February 2012.
- [5] J. Sponsler, HPARSER: Extracting formal patient data from free text history and physical reports using natural language processing software. *Proceedings of the American Medical Informatics Association Annual Symposium*, Washington DC, Nov 3-11, 2001.
- [6] J Sponsler, MD, M Senta, A Kendrick-Adey. Stroke expert system extended to diagnosis CADASIL. *Proceedings of the IASTED Conference for Telehealth*, Innsbruck Austria, February 2012.
- [7] J. Sponsler, F. VanScoy, J. Culberson, The System for Neurological Analysis of Patient Symptoms: An Integrated Artificial Intelligence Prototype. *Proceedings of the IASTED International Conference on Modelling, Simulation and Optimization (MSO2004)*, August 17 -19, 2004, in Kauai, Hawaii.
- [8] G. Steele, *Common Lisp the Language*, 2nd Edition, Digital Press, 1990.
- [9] S. Keene, 1989. *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*, Addison Wesley, 1989, ISBN: 0201175894.
- [10] W.F. Clocksin, C.S. Mellish, *Programming in Prolog Third Edition*, Springer-Verlag, New York 1987.
- [11] J.A. Robinson, A Machine-Oriented Logic Based on the Resolution Principle, *Journal of the ACM* 12:23-44, January 1965.
- [12] Sponsler, J, F Van Scoy, D Pacurari, 2002. An object oriented neuroanatomic atlas, *Proceedings of the Conference of Artificial Intelligence and Soft Computing*, Banff Canada, July 2002.
- [13] UMLS Reference Manual, 2011. National Library of Medicine, Bethesda, MD.
- [14] Sponsler J. 2013. PLEXBASE: A digital model of the brachial plexus. *Proceedings of the 24th IASTED Conference on Modeling and Simulation*, Banff, Canada, July 17-19, 2013.
- [15] M. Lucks, and I. Gladwell, 1993. A Knowledge Representation for Multiple Criteria Assessment Problems, in *Proceedings of CAIA-93, the Ninth IEEE Conference on Artificial Intelligence for Applications*, Orlando, Florida, March 1-5, 1993, pp. 100-105.
- [16] M. Stickel, R. Waldinger, M. Lowry, T. Pressburger, and I. Underwood. Deductive composition of astronomical software from subroutine libraries. *Proceedings of the Twelfth International Conference on Automated Deduction (CADE-12)*, Nancy, France, June 1994, 341-355.
- [17] J. Sponsler, M. Werz, M. Kahrman, 2004. Video EEG Expert System: Software to compute seizure focus lateralization and localization prior to epilepsy surgery. *Proceedings of the IASTED International Conference on Modelling, Simulation and Optimization (MSO2004)*, August 17 -19, 2004, in Kauai, Hawaii.
- [18] Sponsler, J. Automated Analysis of Electromyography Data. *Proceedings of the IASTED Conference for Telehealth*, Innsbruck Austria, February 2012.
- [19] J. Sponsler, F. Pan, 2011. An electronic medical record for neurology. *Proceedings of IASTED Conference on Telehealth*, Feb 2012.
- [20] E. Shortliffe, *MYCIN: Computer-Based Medical Consultations*. New York: American Elsevier, 1976.
- [21] P. Brazis, J. Masdeu, J. Biller. *Localization in Clinical Neurology* / Edition 5. Lippincott Williams & Wilkins, Sept 2006.
- [22] PHP Documentation Group, 2013. *PHP Manual*, <http://www.php.net/manual/en/copyright.php>.
- [23] *MySQL 5.6 Reference Manual* <http://dev.mysql.com/doc/refman/5.6/en/>.

[24] *APACHE Reference* <http://httpd.apache.org/>.

[26] Sponsler, J, 2013. StrokeDx: A stroke diagnosis program, *Proceedings of the 24th IASTED Conference on Modeling and Simulation*, Banff, Canada, July 17-19, 2013.

[27] Sponsler, J, 2013. PLEXBASE: A digital model of the brachial plexus, *Proceedings of the 24th IASTED*

Conference on Modeling and Simulation, Banff, Canada, July 17-19, 2013.

[28] P Dimond, 2013. Is brain mapping ready for big science? *Genetic Engineering & Biotech News* Vol 33, No 9 May 2013.