STROKEDX: A LOGIC PROGRAMMING SYSTEM TO DIAGNOSE STROKE

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

ABSTRACT
BACKGROUND: Decision support for stroke diagnosis is important given the complexity of diagnosis. GOAL: Write a software system to diagnose stroke using logic programming technology. METHODS: Using the conventional data elements for neurological workup, a set of benchmarks for stroke syndromes and a peripheral neuropathy syndrome have been encoded. Using logic programming and back-chaining rule base, a program called *StrokeDx* to diagnosis stroke has been created. Development of *Stroke Benchmarks* and rule sets was done at the same time using rapid prototyping. Each diagnostic rule base was applied to all benchmark datasets to compute a confidence factor from 0 to 1. Diagnoses included frontal stroke, occipital stroke, Wallenberg syndrome, and radial neuropathy. RESULTS: The sensitivity of each diagnostic rule set (for the appropriate diagnosis) was 100%. CONCLUSIONS: Our approach has demonstrated technical feasibility, the goals were met, and encoding of other neurological syndromes is in progress.

KEY WORDS
EXPERT SYSTEM, STROKE, PROLOG, ARTIFICIAL INTELLIGENCE

1. Introduction

Stroke is the number three cause of mortality and a major source of disability in the US [1]. Ischemic stroke is a focal area of brain damage caused by insufficient blood flow to a brain region and usually is due to occlusion of a cerebral artery [2]. Computerized stroke diagnosis can support training of medical students and residents and provide aid for non-neurologists in the acute care setting. In this report we describe an AI program StrokeDx that computes stroke diagnoses from patient data using logic programming techniques.

A previous effort [3] was able to compute stroke anatomic localization using an object oriented technology brain atlas; diagnosis was by methods and heuristics. The current system has accomplished major goals: creation of a set of stroke case benchmarks and a set of hypothesis rule sets for stroke anatomic localization; data transfer from a production electronic medical record into expert system; and data transfer from natural language parsing system into expert system. Other AI work in the AK Brain Center lab included a natural language processing system HPARSER [4]. This system parsed free text patient reports and stored data into a database.

Literature Review

Several AI systems have been constructed in past years for stroke diagnosis. The Anatomic Localizer System [5] employed a decision tree algorithm to compute stroke anatomic locations, contained 30 brain sites, and was similar in performance to human experts. A companion system, Mechanism of Stroke Deducer (MOS) contained knowledge of six stroke types and had 65% accuracy of diagnosis when applied to a patient population [6]. A parent system, MAEISTRO [7] provided a superstructure and user interface for stroke diagnosis and employed an Anatomic Localizer and MOS. A program Computerized Medical Decision Making (CMD) using Mount Sinai algorithms produced positive predictive value of 95% for ischemic stroke [8]. A program MICROSTROKE diagnosed stroke and was correct in 72.8% of 250 cases in the Hamburg Stroke Data Bank [9].

Stroke Diagnosis Fundamentals

StrokeDx was designed to emulate the behavior of a neurologist. The clinical analytical process of a neurologist proceeds in this manner: historical information about the symptoms (weakness of the right arm) is obtained. A review of pre-existing conditions (e.g., hypertension) is done. A neurological examination is performed ("strength of right grip is diminished severely"). These data are used to select one (or more) parts of the central nervous system (CNS) or the peripheral nervous system (PNS) where a lesion/disorder might produce these findings. Diagnostic possibilities (usually multiple) are enumerated (stroke, multiple sclerosis, or tumor). Ancillary data (labs, imaging studies) contribute to diagnostic accuracy. MRI diffusion restriction, CT hypodensity, and Babinski are stroke signs [2].

Prototype Goals

Goals were enumerated for this research effort: (1) Create a set of benchmark datasets for several nervous system disorders (stroke and radial nerve palsy). (2) Create rule trees for stroke syndromes, and radial neuropathy. (3) Test system components (rules against benchmarks) and report on results. (4) Transfer patient cases from EMR to database and test stroke rules. (5) Augment natural language parser with a module to scan text to pick a focused grammar set. (6) Write new grammar rules and constraints to process free text reports for the benchmark stroke cases; transfer data automatically from the reports to StrokeDx and analyze using StrokeDx rules.

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. Descriptions of these medical diagnoses can be found in Brazis [2]. Benchmark data includes patient background, examination findings, magnetic resonance imaging (MRI) findings, computed tomography (CT) findings, magnetic resonance angiogram (MRA) findings, and computed tomography angiogram (CTA) studies.

The system uses these engineering tools: Common Lisp [10], Common Lisp Object System [11], PROLISP [12]. 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"). A Prolisp fact is a pattern (similar to prolog facts). Methods to convert patient pxdata information to Prolisp facts were written.

The benchmark cases encode patient demographics, stroke risk factors, examination data (e.g., weak right biceps), CT image analysis (hypodensity left frontal), MRI image analysis (diffusion restriction left frontal), CTA results (occlusion of left middle cerebral artery), MRA results (occlusion of left middle cerebral artery).

Frontal Stroke Benchmark. The benchmark for frontal lobe ischemic stroke includes weakness of muscles of the contralateral (CL) arm, weakness of muscles of CL face, CT hypodensity in the ipsilateral (IL) frontal lobe, MRI diffusion restriction IL frontal lobe, CTA occlusion IL middle cerebral artery, MRA occlusion IL middle cerebral artery.

Occipital Stroke Benchmark. The benchmark for occipital stroke includes CL visual field deficit, CT hypodensity in the IL occipital lobe, MRI diffusion restriction IL occipital lobe, CTA occlusion IL posterior cerebral artery, MRA occlusion IL posterior cerebral artery.

Wallenberg Benchmark. The benchmark for Wallenberg Syndrome includes IL loss of facial sensation (pain and temperature), CL loss of pain and temperature sensation of arm and leg, IL CT hypodensity in lateral medulla, IL MRI diffusion restriction in lateral medulla, and CTA and/or MRA occlusion of IL posterior inferior cerebellar artery.

Radial Neuropathy Benchmark. A benchmark for radial neuropathy (associated with weakness of IL wrist extensors, normal brain images, and normal angiogram) was encoded. Arm weakness is common to ischemic stroke (a central nervous system CNS condition) and radial neuropathy (a peripheral nervous system PNS condition). StrokeDx differentiation between CNS and PNS disorders was a project goal.

Electronic Medical Record. We have developed an electronic medical record (NEMR) database management system for a neurology clinic using conventional tools [13]. An interface between NEMR has been developed to transfer patient data to the expert system described in this report. The interface populates the main benchmark data structure.

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 [14]. A mathematical operator, *alpha*, is employed in this system [15]. Applied to confidence factors, alpha combines values synergistically.

2. Utilization of logic programming tool PROLISP.

In a previous medical AI experiment, a custom built *Prolog in Lisp* system called **Prolisp** was developed [11]. Prolisp uses CLOS classes to store rules and facts. Prolisp uses *facts*, *rules*, *unification*, and *resolution* in a manner similar to Prolog [16, 17]. Prolisp variables have a question mark prefix (e.g., ?var). 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. Prolisp also incorporates programming concepts from a system called SNARK [18]. 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. 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)
```

Lisp function *proof* runs the Prolisp search.

```
(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.

Prolisp Default Rules

A Prolisp proof will fail if appropriate facts or rules are not found. A prover 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. An example follows and includes a clause to assign default confidence (usually 0.5). Meiosis describes an abnormally small pupil.

```
  (define-rule
    '(HAS-MEIOSIS ?side ?cf)
    '((set-default-cf ?cf)))
```

In the case where the principle query for has-meiosis fails, this default rule will succeed binding the ?cf variable with “unknown” value of 0.5.

3. Benchmark to Prolisp Facts

The initial step is to execute code that takes the benchmark dataset (a CLOS object) and for each attribute create a Prolisp fact. A subset of the fact base (typical count is nearly 100) 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.

```
(CTA POSTERIOR-CEREBRAL-ARTERY :LEFT OPEN 1.0)
(CTA POSTERIOR-CEREBRAL-ARTERY :RIGHT
OCCLUDED 1.0)
(IPSILATERAL :LEFT :LEFT)
(IPSILATERAL :RIGHT :RIGHT)
(BABINSKI :LEFT YES 1.0)
(BABINSKI :RIGHT NO 1.0)
(VISION HOMONYMOUS-HEMIANOPSIA :LEFT 1.0)
(VISION HOMONYMOUS-HEMIANOPSIA :RIGHT 0.0)
(CONTRALATERAL :LEFT :RIGHT)
(CONTRALATERAL :RIGHT :LEFT)
(MRA POSTERIOR-CEREBRAL-ARTERY :LEFT OPEN 1.0)
(MRA POSTERIOR-CEREBRAL-ARTERY :RIGHT
OCCLUDED 1.0)
```

Example Prolisp rule: Wallenberg Syndrome

The rule for diagnosis Horner Syndrome is stated here. Three exam findings are required (ptosis, meiosis, and anhidrosis) and the confidence factors are combined using the alpha operator. Ptosis is eyelid droop, meiosis is abnormally small pupil, and anhidrosis is warm/dry face.

```
(pro:define-rule
 '(horner-syndrome ?side ?cf)
 '((has-ptosis ?side ?p-cf)
  (has-meiosis ?side ?m-cf)
  (has-anhidrosis face :right ?a-cf)
  (alpha3 ?p-cf ?m-cf ?a-cf ?cf)))
```

The top level rule for diagnosis Wallenberg is stated here in an abridged version.

```
(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)
  (loss-of-pain-sensation body ?cl-side ?body-pain-cf)
  (loss-of-temperature-sensation body ?cl-side ?body-temp-
cf)
  (average6 ?horner-cf ?ataxia-cf ?pain-cf ?temp-cf ?body-
pain-cf ?body-temp-cf ?wallenberg-cf)))
```

Rule Example: Occipital Stroke

The semantics of this rule: An occipital stroke is diagnosed if there is a contralateral vision deficit and CT shows ipsilateral hypodensity and MRI shows ipsilateral diffusion restriction and angiogram shows ipsilateral PCA occlusion and there are stroke risk factors.

```
(define-rule
 '(OCCIPITAL-STROKE ?side ?stroke-cf)
  '((contralateral ?side ?cl-side)
   (= ?lobe occipital)
   (visual-fields homonymous-hemianopsia ?cl-side ?vf-cf
?vf-trace)
   (ct-hypodensity ?lobe ?side ?ct-cf ?ct-trace)
   (mri-dwi-positive occipital ?side ?diffusion-cf ?dwi-trace)
```

```
  (stroke-risk-factors ?risk-cf ?risk-hx)
  (angiogram-occlusion posterior-cerebral-artery ?side
?angiogram-cf ?angiogram-hx)
  (average5 ?vf-cf ?ct-cf ?diffusion-cf ?risk-cf ?angiogram-cf
?stroke-cf)))
```

Proof yields confidence factor and explanation tree.

The *proof* operator tests the diagnosis and binds the *?cf variable* (the computed confidence) and *?trace variable* (not shown in previous examples). The ?trace variable provides a trace of the depth first search through rule space which produced the proved solution. The ?trace data is in lisp format and so can be printed in a clear manner for diagnostic explanation.

```
(proof (occipital-stroke :right ?cf ?trace))
?CF = 1.0
?TRACE = (STROKE (LOBE OCCIPITAL) (SIDE :RIGHT)
(CF 1.0) (HOMONYMOUS-HEMIANOPSIA (SIDE :LEFT)
(CF 1.0))   (CT-HYPODENSITY (LOBE OCCIPITAL) (SIDE
:RIGHT) 1.0)   (MRI-DIFFUSION-POSITIVE (LOBE
OCCIPITAL) (SIDE :RIGHT) 1.0) (ANGIOGRAM-
OCCLUSION  (ANGIOGRAM-OCCLUSION (ARTERY
POSTERIOR-CEREBRAL-ARTERY) (SIDE :RIGHT) (CF
1.0)  (CTA (CT-ANGIOGRAM-OCCLUSION (ARTERY
POSTERIOR-CEREBRAL-ARTERY) (SIDE :RIGHT) (CF
1.0)))  (MRA (MR-ANGIOGRAM-OCCLUSION (ARTERY
POSTERIOR-CEREBRAL-ARTERY) (SIDE :RIGHT) (CF
1.0)))  (COMBINER ALPHA)))
```

Test Results (Benchmark vs Diagnosis)

The testing algorithm: Loop over each benchmark and apply all stroke rules to that benchmark. Collect confidence factors.

For all benchmarks, the correct diagnosis was found (had the CF closest to 1). For a benchmark, the contralateral diagnosis (left frontal vs right frontal) calculated CF was found to be correctly near zero. This is because the benchmark would have all findings (positive or negative) required by the specific diagnosis and no default values.

For certain diagnoses (RT Frontal) applied to another brain locus (LT Occipital), required facts were more likely to not have been encoded and thus to have defaulted to unknown (CF 0.5). The confidence in that benchmark/diagnosis pair then tends to have combined confidence of 0.5. After analysis revealed the CF 0.5 results, benchmark files were augmented with the missing facts (normal patient values) and factors were then closer to the correct zero values.

BENCH:	**F LT**	**F RT**	**O LT**	**O RT**	**W LT**	**W RT**	**R LT**	**R RT**
Dx F LT	**1.0**	0.17	0.35	0.33	0.17	0.17	0.17	0.17
Dx F RT	0.25	**1.0**	0.33	0.5	0.33	0.33	0.17	0.17
Dx O LT	0.2	0.2	**1.0**	0.2	0.2	0.2	0.2	0.2
Dx O RT	0.4	0.2	0.2	**1.0**	0.2	0.2	0.2	0.2
Dx W LT	0.0	0.0	0.0	0.0	**0.83**	0.0	0.0	0.0
Dx W RT	0.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0
Dx R LT	0.5	0.5	0.75	0.25	0.75	0.75	**0.94**	0.56
Dx R RT	0.5	0.5	0.25	0.75	0.75	0.75	0.56	**0.94**

Table 1. Benchmark vs. Diagnoses. The benchmarks are columns and diagnoses are rows. Each table cell is a diagnostic confidence factor. For a given benchmark (e.g., F LT) the CF of each diagnosis is given. In this table, F = frontal, O = occipital, W = Wallenberg, R = radial palsy. LT is left and RT is right. Recall that CF 0 is false, CF 0.5 is unknown, and CF 1 is true.

EMR TEST CASE:	LT FRONTAL	RT OCCIPITAL	RT RADIAL
Dx F LT	**0.75**	0.33	0.4
Dx F RT	0.33	0.33	0.4
Dx O LT	0.4	0.4	0.3
Dx O RT	0.4	**0.7**	0.4
Dx W LT	0.5	0.5	0.5
Dx W RT	0.5	0.5	0.5
Dx R LT	0.45	0.25	0.6
Dx R RT	0.31	0.5	**0.66**

Table 2. Hand generated database cases with StrokeDx confidence factors. The RT Occipital/R LT pair value of 0.25 reflects that the CT showed stroke which counts against radial neuropathy. For each test case, StrokeDx gave the correct diagnosis the highest relative score (bold numbers).

Database Test Cases

A set of test cases were created using the NEMR database. These cases were not actual patients but instead were hand constructed cases. The cases included frontal stroke, occipital stroke, and radial neuropathy. Interface files were generated by command and these files were then tested using StrokeDx. Results are in Table 2. Based on confidence factor sorting, StrokeDx resulted in the correct diagnosis for each case. These cases differ from the benchmarks cases: the data sets are incomplete with many data elements missing. This forced StrokeDx to use default CF 0.5 which pulls diagnosis CF toward the unknown confidence.

4. Natural Language Parser for Stroke Cases

In a previous research effort [3], we described SYNAPS a diagnostic engine that used a network of CNS objects (a digital neuroanatomic atlas). The front-end for SYNAPS, HPARSER [4], 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 is satisfied (or not) and data found in the tree is collected and stored. 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.

HPARSER/StrokeDX Results

In the current effort, patient free text reports were created for frontal stroke, occipital stroke, Wallenberg stroke, and radial neuropathy. A file might contain a 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 listed in Table 3. 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).

PARSER TEST CASE:	RT FRONTAL	RT OCCIPITAL	RT RADIAL
Dx F RT	**0.75**	0.33	0.33
Dx F LT	0.42	0.42	0.33
Dx O LT	0.4	0.4	0.3
Dx O RT	0.4	**0.6**	0.4
Dx W LT	0.5	0.5	0.5
Dx W RT	0.5	0.5	0.5
Dx R LT	0.31	0.18	0.43
Dx R RT	0.44	0.56	**0.56**

Table 3. Free Text Stroke Diagnosis. Fabricated parser cases with StrokeDx confidence factors. For each test case, StrokeDx computed the highest relative score (bold numbers) for the correct diagnosis.

5. Conclusions

The following conclusions are made from this research effort:

1. This prototype has demonstrated the utility of creating benchmarks containing positive and negative patient signs and the benchmarks can then exercise diagnostic systems. A large library of benchmarks is the goal of future work.

2. StrokeDX (a logic programming expert system) is a powerful diagnostic system with high sensitivity. 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. The default rules support rule set search completion and so missing data does not cause search to fail. The default value of 0.5 (unknown) semantically is reasonable insofar as the number does not support or deny a diagnosis.

3. Interface program was able to transfer data from Neurology Electronic Medical Record (NEMR) to the StrokeDx examination data structure. Diagnoses were correct for the research test cases.

4. The revised HPARSER NLP prototype demonstrated facility in parsing free text patient reports efficiently, transferring data (based on grammar rule constraints) to the StrokeDx examination data structure. Test cases were correctly diagnosed. HPARSER was evolved to support partitioned grammar sets that are selected by keyword sentence scanning. This change resulted in improved speed and parsing precision.

Future work includes creation of benchmark datasets and diagnostic rule sets for other neurological syndromes such as lacunar stroke syndromes, carpal tunnel syndrome, multiple sclerosis, brachial plexopathy, and others. Based on the experience reported above, development of benchmarks and rule sets will be relative synchronous. Testing the diagnostic engine against real patient data is also planned.

References

[1] U.S. Centers for Disease Control and Prevention and the Heart Disease and Stroke Statistics - 2007 Update, published by the American Heart Association. http://www.strokecenter.org/patients/stats.htm.

[2] P. Brazis, J. Masdeu, J. Biller. *Localization in Clinical Neurology* / Edition 5. Lippincott Williams & Wilkins, Sept 2006.

[3] 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.

[4] 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.

[5] G. Atkinson, D. Hier, W. McCormick, Development, refinement, and evaluation of a knowledge-base for a stroke expert system. *Proc. of the 9th Annual Symp. on Computer Applications to Medical Care*, Baltimore MD, pp. 320-324, Nov 10-13, 1985.

[6] B. Desai, M. Beachy, K. Biss, M. Evens, D. Hier, H. Hill, W. McCormick, Comparing Bayes rule and the nearest neighbor rule for medical diagnosis. *Proc. 1985 Conf. on Intelligent Systems and Machines*, Rochester MI, Apr 22-24, 1985.

[7] D. Hier D, MAIESTRO: The Michael Reese Hospital - Illinois Institute of Technology Expert System for Stroke, *Proc Annu Symp Comput Appl Med Care*. 1990 November 7: 995–996.

[8] R. Wain, S. Tuhrim, L. D'Autrechy, and J.A. Reggia. *Proc Annu Symp Comput Appl Med Care*. 1991: 94–98. The design and automated testing of an expert system for the differential diagnosis of acute stroke.

[9] K. Spitzer, A. Thie, L. Caplan, K. Kunze, The MICROSTROKE expert system for stroke type diagnosis. *Stroke*. 1989 Oct;20(10):1353-6.

[10] G. Steele, *Common Lisp the Language, 2nd Edition*, Digital Press, 1990.

[11] S. Keene, 1989. *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*, Addison Wesley, 1989, ISBN: 0201175894.

[12] J. Sponsler, M. Werz, M. Kahriman, 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.

[13] J. Sponsler, F. Pan, 2011. An electronic medical record for neurology. *Proceedings of IASTED Conference on Telehealth*, Feb 2012.

[14] E. Shortliffe, *MYCIN: Computer-Based Medical Consultations*. New York: American Elsevier, 1976.

[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] W.F. Clocksin, C.S. Mellish, *Programming in Prolog Third Edition*, Springer-Verlag, New York 1987.

[17] J.A. Robinson, A Machine-Oriented Logic Based on the Resolution Principle, *Journal of the ACM* 12:23-44, January 1965.

[18] 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.