Kenyan "Nyumba Kumi" Neighborhood Information System using Artificial Intelligence

Kevin O. Gogo
Chuka University
P.O box 109,
60400 Chuka - Kenya

Kenneth O. Sigar
Chuka University
P.O box 109,
60400 Chuka - Kenya

ABSTRACT
On our daily life, we need to engage with our neighbors on several issues ranging from social, security and general neighborhood wellbeing. In this regard, we have designed a "nyumba kumi" information system which uses artificial intelligence and the facts stored in its knowledge base to answer various "nyumba kumi" neighborhood queries. We collected a sample "nyumba kumi" concerns and converted them into facts and the system rules. Hence, we fed the system with sample facts converted into predicate logics, which we collected from a certain "nyumba kumi" neighborhood. The system is able to answer various neighborhood queries using artificial intelligence knowledge from the facts and rules fed into the system, which we designed using prolog. The system could be customized by adding as many facts and rules as possible as long as the facts and rules are not contradicting.

General Terms
Artificial intelligence, "nyumba kumi" (ten homes),

Keywords
Artificial intelligence, prolog, neighborhood information system, "nyumba kumi"

1. INTRODUCTION
On our daily life, we need to engage with our neighbors, either on security issues, family relations, social issues etc. The engagement with our neighbors enables our society to be consistent and upright in the decisions and social wellbeing of its members. This information system is a system that has been given the various rules pertaining to the neighborhood, and from those rules it should is able to deduce some social neighborhood issues and give appropriate solutions concerning our neighborhood queries. The details of members the "nyumba kumi" (ten homes) are written in prolog and fed into the system. Hence from the facts and rules that have been fed in the system using prolog, we can now be able to get the information of members of the "nyumba kumi" when queried from the system.

1.1 Problem statement
The primary goal of this study is to design and build a Kenyan "nyumba kumi" neighborhood information system using prolog, then implement, test and evaluate the system. The system is built with a view to return the "nyumba kumi" neighborhood members details using the artificial intelligence, based on the facts and rules defined in the system. The facts and rules of "nyumba kumi" neighborhood members should be able to be added on the move, in order to make the system more knowledgeable on "nyumba kumi" neighborhood issues.

1.2 Research objectives
The research objectives of this project are as follows:-

- To design and build a Kenyan "nyumba kumi" neighborhood information system using prolog
- To implement, test, and evaluate the system with some neighborhood details

2. RELATED SOLUTIONS AND LITERATURE REVIEW
In the literature review, we reviewed some systems which have been built using prolog, and the importance of using prolog in doing the same.

2.1 Related solutions
Reference [3], [8], [13] and [14] Explains that there are quite a number of related projects tackled using prolog and other artificial intelligence systems, which are in line with our information system, in the field of computing. Some of the related solutions include: -

2.1.1 Family relations using artificial intelligence
This prolog system was designed to give the family relations, such that I would key in all the members of an extended family with their relations and characteristics such as male, female, parent of etc. the prolog would then use the artificial intelligence knowledge and provide for their relations such as siblings, grandchildren, spouse etc

2.1.2 Social relations using artificial intelligence
Using prolog a social relations system has been developed. The social relations system is a system that could be able to determine the social activity one would be able to participate in given a number of person characteristics and likes [14].

2.2 Our contribution
We are contributing the design of the "nyumba kumi" (*ten homes*) information system using a prolog artificial intelligence system. Kenya and its Tanzanian counterpart is still in the process of designing a system that will be able to capture the information regarding the "nyumba kumi" which is not yet underway

2.3 Literature review
2.3.1 "Nyumba kumi"
"Nyumba kumi" is a Swahili word for ten homes. This is a phenomenon being brought about in Kenya. It is a way in which ten homes within the neighborhood should come together and keep watch of their neighbors, especially on issue to do with neighbor's wellbeing, behaviors, characters and security matters. If the "nyumba kumi" initiative is rolled successfully we expect neighbors to know each other well and

to improve on their security through keeping an eye on the other, for safety reasons, [7].

2.3.2 Artificial intelligence

Artificial intelligence, as pointed by [13] as an area of computer science with the design of intelligent information system, with systems inhibiting characteristics of human experts. Artificial intelligence could be in the area of knowledge based, speech recognition, natural language processing, robotics, automatic programming, perceptions etc.

Reference [9] Further clarifies that artificial intelligence has brought emergence of a number of technologies which enable the computers to exhibit human thinking and reason how an expert would reason, hence it is the way forward for systems.

2.3.3 Artificial intelligence programming

Artificial intelligence programming for knowledge based systems entails some formalism as viewed with [5] and [9]. The formalism for our programming include: -

Production rules: in which the database of knowledge consist of rules in the form of condition action pair "if this condition occur then do this action"

Structured objects: in which a structure consists of nodes representing concepts and objects and links between the objects representing relationships

Predicate logic: in which information is represented using formal logics, allowing facts to be deduced using the rules of inferences

They further clarify that there are several artificial intelligence tools ranging from prolog, COBOL, Weka, etc.

2.3.4 Prolog

Arguments from [11] and [2] points out that Prolog is a symbolic artificial intelligence programming tool particularly suited to distributed applications unlike other work on distributed logic programming. It does not aim to improve the raw performance of a logic program nor require multiprocessor machines or specialized hardware. Instead it widen the applicability of logic programming to encompass new classes of practical applications which require the co-ordination of concurrently executing programs on separate workstations to communicate over the network. Prolog also enables concurrent execution of independent goals with a language well-suited for writing network-friendly applications

2.3.5 Benefits of prolog in Artificial intelligence programming

[12] and [2] further advices that prolog as compared to other artificial intelligence programming tools have some benefits which are not limited to: -

Prolog do not require bigger computer memories, they use logics which are simple to design, and with goals clearly defined, adhoc queries, pattern matching, low cost, takes less CPU time, as well as they can perform concurrent executions.

3. METHODOLOGY

3.1 Systems design

In system designs as advised by [11] and [1], we developed family and social interaction rules of the neighborhood from the quick interview of some issues that bind people within the neighborhood. These issues were converted as predicate logic equivalents using the artificial intelligent knowledge, which were then fed into prolog. From the neighborhood data we developed the rules, which were either general family rules or society rules. We then populated both the facts and the rules pertaining to the neighborhood in prolog, and developed a neighborhood information system. We also gave some standard list of predicates that should be used and a way of coming up with predicates which were to be in line with the system.

3.2 Experimental and data sources design

Following [10] procedures we developed the "nyumba kumi" information system using prolog, thus having keyed in some facts from a randomly selected "nyumba kumi" combined with the rules we developed for the "nyumba kumi". We then experiment the system through testing the intelligence of the system by asking queries pertaining to the facts that we keyed in, and made the records of the solutions received with the various deductions of the system.

3.3 Data collections procedures, methods and tools

In data collections as argued by [6] in our information system we collected the data through experiments. We recorded the results as the system was being tested. On the other hand, the data that enabled us to come up with the systems predicates for the "nyumba kumi" neighborhood were gathered from a quick interview that we conducted on the residents of Jua-kali neighborhood on the issues of importance that they would like to be kept concerning the "nyumba kumi" neighborhood.

3.4 System implementation

Taking advice from [10] and [3], in the implementation of this system, first we interpreted the concerns of "Nyumba kumi" and converted them into prolog equivalents, in form of predicates and logics. On the predicate part we used agreed predicates in simple past tenses of each and every concern that was to be stored with the system i.e. if otieno lives in plot No.1, we represented as *lives(otieno, plot_no1)*, if njoroge stolen a cow, we represented as *stolen(njoroge, cow)* etc. some of the agreed predicates include male, female, married, fought, stolen, bewitched, born, jailed, drink and as many as the neighborhood concerns. From the agreed predicates we developed the facts and implemented them in the system.

We also developed the rules which were to ensure that the repetitions were minimized and some synonyms used i.e.

```
grandparent(X, Z)      :- parent(X, Y), parent(Y, Z).
thief(X)               :- stolen(X,Y).
witch(Y)               :- bewitched(Y,X).
murderer(Y)            :- killed(Y,X), is(X, person).
warrior(X)             :- fought(X,Y).
etc
```

As the neighborhood concerns keep developing as says go by, we expect that these rules are not exhaustive, but will keep developing as more facts are being added in the system, with a view to expanding the system knowledge base.

3.5 Testing and results

We conducted the following tests to the system: -

Unit testing: as we were developing the units we tested the units separately, i.e. as we include the facts we tested to confirm that the facts were correct. The same applied to the rules that we were developing.

System testing: after we had integrated all the parts we tested the system as a whole to confirm its correctness, as well as with the various data/facts keyed in.

4. IMPLEMENTATION

4.1 Sample facts fed into the system

Using advice from [4] there were several facts that were received from the selected "nyumba kumi" and converted into prolog equivalent logics. A number of those facts included: -

```
female(mbula).
female(mercy).

parent(otieno, njoroge).
parent(kimani, wanjiru).

married(mbula, baraza).
married(nancy, john).

born(kimani, 1983).
born(kanini, 1978)

fought(wamboi, kanini).
fought(nancy, john).

killed(mbula,baraza).
killed(nancy,munene).

lives(otieno,plot_no1).
lives(kimani,plot_no1).

bewitched(pam,nancy).
bewitched(baraza,mbula).

jailed(kimani,industrial_area).
jailed(otieno,kamiti)

drink(kimani, beer).
drink(mwala, spirits).

drink(ian, wine).
```

4.2 Sample rules for the "nyumba kumi"

There were a number of rules that we developed. A sample of these rules includes the following (converted into predicate logic equivalents): -

4.2.1 Family relation rules

Some of the family relation rules that are guiding the artificial intelligence system are as follows [13] and [14]: -

```
spouse(X, Y)        :-  married(X, Y).
husband(X, Y)       :-  male(X),   married(X, Y).
wife(X, Y)          :-  female(X),    married(X, Y).
father(X, Y)        :-  male(X),      parent(X, Y).
mother(X, Y)        :-  female(X),    parent(X, Y).
child(Y, X)         :-  parent(X, Y).
son(Y, X)           :-  male(Y),      child(Y, X).
daughter(Y, X)      :-  female(Y),    child(Y, X).
sibling(X, Y)       :-  father(Z, X),  father(Z, Y),
                        mother(W, X),  mother(W, Y),    not(X = Y).
```

4.2.2 Social relation rules

Some of the social relation rules that are guiding the artificial intelligence system are as follows: -

```
thief(X)            :-stolen(X,Y).

witch(Y)            :- bewitched(Y,X).

murderer(Y)         :-killed(Y,X). is(X=person).

stay(X,Y)           :- lives(X,Y).

warrior(X)          :- fought(X,Y).

drunkard(X)         :- drink(X,Y).
```

5. TESTING AND RESULTS

5.1 testing the system with no query given

When the system was opened and no query typed on the system, it displayed a blank prolog screen as shown in fig 1.

```
% c:/Users/friend/Desktop/nyumba Kumi.pl compiled 0.02 sec, 450
clauses

Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.6.6)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is
free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
```

Fig 1: Testing the system without any query typed on the system

5.2 Testing the system asking some queries based on neighborhood relation issues

The system returned the following solutions, as shown in figure 2, when some queries were keyed on it

```
For help, use ?- help(Topic). or ?- apropos(Word).
1 ?- female(X).
X = wamboi ;
X = akinyi ;
X = wanjiru ;
X = joan .

2 ?- parent(otieno, X).
X = njoroge.

3 ?- lives(otieno, X).
X = plot_no1.

4 ?- lives(X, plot_no1).
X = otieno ;
X = kimani.

5 ?- thief(X).
X = njoroge ;
X = otieno.

6 ?-
```

Fig 2: Testing the system on some neighborhood related issues

5.2.1Query 1, female in the neighborhood

It returned wamboi, akinyi, wanjiru among others

5.2.2Query 2, parent of otieno

It returned njoroge

5.2.3Query 3, where do otieno lives

It returned plot_no1

5.2.4Query 4, female in the neighborhood
It returned wamboi, akinyi, wanjiru among others

5.3 Testing the system asking more queries based on the situational rules

The system returned the following solutions, as shown in figure 3, when the alongside queries were keyed onto the system.

```
6? - stolen(njoroge, X).
X = chicken.

7? - stolen(otieno, X).
X = cow

8? - witch(X).
X = pam.

9? - bewitched(pam,X).
X = nancy.

10? - drunkard(X).
X = wamboi.
X = mwangi.
X = mwala.     (etc)

11?-
```

Fig 3: Testing the system on further neighborhood related situational issues

5.3.1 Query 6, what did njoroge stole
It returned chiken

5.3.2 Query 7, what did otieno stole
It returned cow

5.3.3 Query 8, list witches in the neighborhood
It returned pam etc

5.3.4 Query 9, who did pam bewitched
It returned nancy

5.3.5 Query10, list the drunkards in the neighborhood
It returned wambui, mwangi, mwala, ian, otieno etc

5.4 Testing the system on further social relations

The system returned the following solutions, as shown in figure 3, when the alongside queries were keyed onto the system.

```
11? - bewitched(X, nancy).
X = pam.

12? - drink(X, beer).
X = kimani.

13? - born(1978).
X = kanini.
        several others also displayed

14? - husband(mbula, X).
X = baraza.


11?-
```

Fig 4: Testing the system on further social relations issues

5.4.1 Query 11, who bewitched nancy
It returned pam

5.4.2 Query 12, who drinks beer
It returned Kimani

5.4.3 Query 13, list persons born in 1978
It returned kanini and several others

5.4.5 Query 14, who is mbula's husband
It returned baraza

5.5 Summary of results displayed by the system

The "nyumba kumi" information system was tested using a various categories of questions whose facts were stored in the knowledge base and the findings tabulated as either correct or incorrect. With the correct answers having displayed all correct solutions and incorrect having displayed either partial correct solutions or fully incorrect solutions pertaining to the "nyumba kumi".

Table 1. Table showing summary of results from the system

Table of results of a nyumba kumi system				
Question category	**Questions tested**	**Correct solution**	**Incorrect solutions**	**% of correct solutions**
Family relations	100	90	10	90%
Social relations	100	78	22	78%
General information	100	64	36	64%
% OF AVERAGE CORRECTNESS				**77.33%**

The system gave correct solutions value of 77.33%, with incorrect or partially correct solutions value of 22.66%. This shows that the system would be beneficial if it's put in place, with up to date knowledge base pertaining to any "nyumba kumi". The system also comes over with the simplicity of implementation and usage. The system also allowed for adhoc queries and addition of knowledge base contents on the go.

6. CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The main objective of this system was to use artificial intelligence to answer queries regarding the ten homes neighborhood, using the facts fed into the system. These facts should be collected from a "nyumba kumi" neighborhood. We designed and implemented the system, and then gave a structure of how to develop the predicate logics for feeding the system. Having fed the system with facts using correct rules, the system was able to answer the queries pertaining to the facts given. I.e. having given gender, name, and parents of a "nyumba kumi" neighborhood member, the system should be able to return the siblings, children and grandparents of the member. The system was also able to answer several other queries pertaining to the "nyumba kumi" neighborhood.

6.2 Recommendations

The system managed to solve and give correct solutions on the questions which were presented to it, with regard to the facts that were filled on the program. We also do recommend the following further developments on the systems: -.

- The system should be upgraded to pick on the other natural languages, such that we do not need to ask the questions in English only. That is the rules should also be expanded to pick other natural languages and synonyms.
- A user friendly interface which interacts with prolog, that allow the filling and asking of the questions without moving to prolog window
- The village using the system should develop a unique identifier for distinguishing every person as there could be more than one person with the same name in the village.
- Several predicate and facts should continue constantly be included for each and every individual in the neighborhood to enable the system capture even the very critical issues of the selected "nyumba kumi" such that very finer details of every individual in a neighborhood should be available in the system.

7. REFERENCES

[1] Bitter, C., Elizondo, D., and Yang, Y,. 2009, Natural language processing: a prolog perspective, Springer science and business media

[2] Brna, P., et al 1991. Prolog programming techniques. Journal of instructional science Volume 20, Issue 2-3 , pp 111-133

[3] Clocksin, W., and Mellish, C., 2003, Projects in Prolog In Programming in Prolog using ISO standards, pp 259-265

[4] Clocksin, W., F., et al 2003, Programming in prolog, Springer Verlag

[5] Getzinger, T., 1994, The costs and benefits of abstract interpretation-driven Prolog optimization, Static Analysis Lecture Notes in Computer Science Volume 864, 1994, pp 1-25

[6] Kothari C.R. 2004. Research methodology, 2nd ed., New Delhi: New Age Int. (P) Ltd.

[7] Nation Television news, 2014, available at http://ntv.nation.co.ke/news2/topheadlines/nyumba-kumi-initiative-gated-communities-have-advanced-security-systems/ accessed on 9th Feb 2016

[8] O'Keefe, R. A. 1990. The craft of Prolog. Cambridge, MA: MIT Press

[9] Pau, L., and Gianotti, C., 1990, Artificial Intelligence Programming Languages In Economic and Financial Knowledge-Based Processing Springer, pp 123-184

[10] Readt, L., and Kimming, A., 2015, probabilistic (logic) programming concepts, Springer

[11] Smedt K., 1998, Knowledge representation techniques in Artificial Intelligence, an overview In Human computer interaction , Springer Verlag

[12] Stickel, E., M. 1986, A prolog technology theorem prover: Implementation by an extended prolog compiler In 8th International Conference on Automated Deduction, Lecture Notes in Computer Science Volume 230, pp 573-587

[13] Treleaven, P., Refenes, A., Lees, K., and McCabe, S., 1987, Computer architectures for artificial intelligence in Future In Parallel Computers, Springer Volume 272, pp 416-49

[14] Wagner, R., N. 2000, knowledge bases and the prolog language, online available at http://www.cs.utsa.edu/~wagner/CS1023/readings/prolog.html