Universal Journal of Educational Research 4(5): 941-948, 2016
DOI: 10.13189/ujer.2016.040502 http://www.hrpub.org

What Will Be in Those Lap Tops: Empowering Students and Teachers to Add Content to an Educational Chatbot's Knowledge Base

Patrick K. Bii[1,*], J. K. Too[2]

[1]Department of Mathematics and Computer Science, University of Kabianga, Kenya
[2]Department of Curriculum Instruction and Educational Media, School of Education, Moi University, Kenya

Abstract The Government of Kenya has in the recent past been on a serious footing to put in place ICT initiatives and projects aimed at realizing the potential within its recognition of ICT as a foundation for a knowledge economy as per vision 2030. The tempo has risen several notches higher with the current intention and initiative of making available laptops to pupils and students in primary schools in the years to come. However, once computer hardware is available to schools as PC's, Laptops and other mobile computing devices, a critical issue which must be adequately addressed is the educational content within. This paper outlines a possible approach using open source software development concepts and the Participatory Action Research (PAR) model as a means of empowering students and teachers to become active developers of the content they consume.

Keywords ICT, Educational Software, Participatory Action Research, Chatbot

1. Continuing Efforts to Supply Computers for Educational Purposes

The Government of Kenya has in the recent past been on a serious footing to put in place ICT initiatives and projects aimed at realizing the potential within its recognition of ICT as a foundation for a knowledge economy as per vision 2030 [1, 2, 3]. The tempo has risen several notches higher with the current intention and initiative of making available laptops to pupils and students in primary schools in the years to come [4, 5, 6, 7]. Realistic cognizance of possible challenges to this initiative (and hence possible ways of surmounting the challenges) is pertinent, for as Anne [8, p1] has noted, "ever since the new government announced in their manifesto that they will be deploying laptops to standard 1 children in schools across Kenya, the big question on everyone's mind is how will this program work? Can the country afford it? What would it take to make the idea work?" Some of the challenges include cost of hardware, cost of software, connectivity, infrastructure, pupil competencies, teacher competencies, school policies, administrative structures and cultures, hardware and software support systems, success metrics, security considerations, educational content, acceptance by teachers and society, danger of misappropriations, sustainability and business and commercial interests [4, 6, 8, 9, 10, 11]. This paper focuses on the critical issue of availing what will be in the laptops and other computers meant for educational use – educational software and content.

2. The Challenge of Availing Educational Software

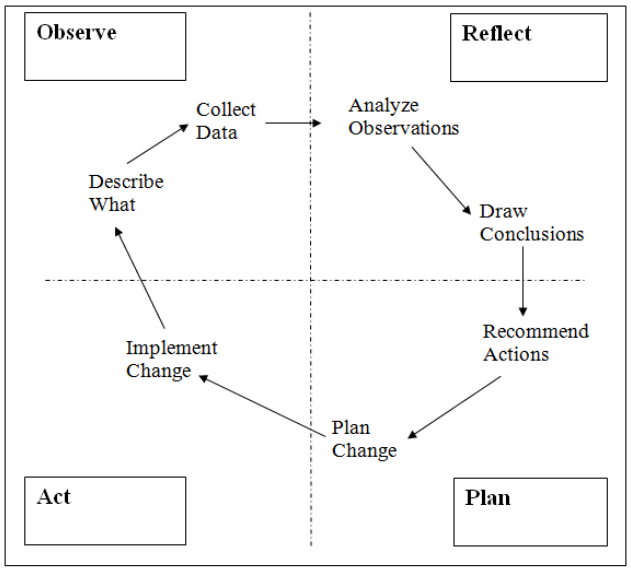
Cautions to be considered in attempts to avail educational software to schools include not seeing the issue in terms of availing hardware alone, what software will be availed, the educational content of the software, scarcity of digitized content relevant to approved curriculum, and development of software that can extract content from school textbooks [6, 8].Further, Hepburn [10, p1] stated that

> An unavoidable part of making ICT available in schools is obtaining and maintaining the software that is necessary to allow school computers to function. Most software that schools use is produced by proprietary software companies that normally charge considerable sums of money for their products. The cost and usage restrictions that characterize proprietary software place an enormous stress on cash strapped schools. As a result of this situation,

> schools are left with a serious problem: they clearly need to integrate ICT into teaching and learning but doing so requires large, ongoing expenditures to purchase and maintain ICT resources.

Hepburn [10] enumerates challenges of using proprietary software as high cost, restrictions on flexibility of use due to licensing constraints, and ethical and social issues including equity and the moral one of exposing students to and training them on particular companies' software while the students pay the proprietary companies to do so.

3. A Suggested Approach to Availing Educational Software Content



Source: 22

Figure 1. Working Participatory Action Research Model

A possible way out for schools and other institutions of learning to the software issue is the use of open source software, software distributed with a license granting access to source code, distribution, modification and free use [12, 13, 14, 10, 11]. Compared to proprietary software, open source software is less costly, offers greater flexibility of use, and is in a position to address the social issues of equity and corporate involvement [10]. This approach has borne fruit in the case of a number of open source software initiatives including the Linux operating system, OpenOffice and LibreOffice, and chatbot ALICE [15, 16, 17, 18]. If a similar approach is made possible for educational software content development, the pool of developers, development, and continuity potentially is unlimited since this approach opens up the software to a large community to become involved in the development effort, allowing rapid bug fixes and enhancements to occur. The potential developers of interest here are students, teachers, lecturers, educational resource persons and other interested developers. The suggested specific content to be developed by this pool of developers is the educational chatbot's knowledge base. This open source approach coupled with the Participatory Action Research model, illustrated in Figure 1, has a possible chance of bearing fruit.

In Participatory Action Research, there is an explicit intention to educate and co-produce change with the collaboration of those affected by the issue being studied [19, 20]. The PAR process involves participants in 'planning action (on the basis of reflection); in implementing these plans in their own action; in observing systematically this process; and in evaluating their actions in the light of evidence as a basis for further planning and action, and so on through a self-reflective spiral' [21, p317]. The steps in practice may overlap and can begin at any point.

4. Adding Content to An Educational Chatbot's Knowledge Base

A chatbot is a computer program that is created to simulate intelligent human language interaction through text or speech and whose purpose is to engage in conversation or to emulate informal chat communication between a human user and a computer using natural language [23, 24, 25]. They can be created using various computer programming languages including PHP, XML, JAVA, C++, Python and AIML. Two approaches can be used in developing a chatbot. One approach is to start with an empty database to which content is automatically added as it is used while the other approach is to have the chatterbot creator program the database so that it has pre-programmed questions, phrases or words and how it is to respond to each question, phrase or word [26]. Chat-logs created during interaction sessions additionally serve as sources for chatbot response improvement [27, 17, 25, 28]. In AIML, the pattern is the user's expected or assumed question (the matching part) while the template is the chatbot's prepared or programmed answer (the returning part). The AIML pattern syntax was designed as a very simple pattern language, but greater complex response can be achieved through use of a few more AIML tags which can, as Wallace [34, p1] states, "transform the reply into a mini-computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories." A further simplification to AIML file creation is possible through use of MakeAiml Editor, created by Dryden [23] and which enables a programmer to reduce the amount of work required to write AIML files through providing a shorthand way to represent AIML tags and their contents [23]. For example, the AIML file

```
<aiml>
<category>
<pattern>HELLO</pattern>
<template>Hi! How are you?</template>
</category>
</aiml>
```

typed in notepad can be typed in more simply as

p Hello
t Hi! How are you?

in MakeAiml. In other words, one need only specify the pattern (p) and template (t) pairs as illustrated above in creating an AIML file.

In order to add content to an educational chatbot's knowledge base, the community of students, teachers, resource persons and interested developers can be tasked to program the database of the educational chatbot with educational content, including questions and answers on school subject content, teaching-learning notes typed into the chatbot directly by students and teachers or sourced from open source e-books, and digital content availed to schools. This has the potential of enhancing interaction and collaboration between students, teachers, resource persons and learning institutions, key requirements of an educational environment that can enable students acquire 21st century skills [29, 30, 31]. Such an approach could be used to develop individual, class, school, regional, national and even international knowledge bases for educational chatbots.

In a specific school, for example, the teacher when teaching a given topic may request students to group themselves according to the number of computers available in the school's computer laboratory or their class, and then program their group's chatbot with content, questions, answers, keywords and definitions concerning the topic being covered. Ideally, the chatbot starts blank without appropriate content, keywords, and responses to questions pertaining to the topic under study. The expectation is that by the end of the programming sessions, the chatbot should be able to respond intelligibly to the teaching-learning topic. The programming sessions could be undertaken at arranged times as per the school's time tabling requirements. Upon completion of chatbot programming, the student group chatbots are rated on comprehensiveness of content and responses made to questions it is asked pertaining to the topic.

5. Knowie: A Concrete Example

A chatbot named Knowie was developed by the present authors with such use in view. The bot's knowledge comes from AIML files. AIML consists of the elements, categories, patterns, and templates. A category is a basic unit of knowledge in AIML. It consists of an input question (pattern or stimulus), an output answer (category or response), and an optional context (topic, that). Implementation software for the technology is open source: Ubuntu Linux, Python, JDK, PyAIML, and MakeAiml. The chatbot Knowie was derived from the open source chatbot Howie, originally created by Stratton [32]. Knowie's basic interface and sample interaction are presented in Figures 2 and 3 respectively. When the user launches the chatbot, it presents clickable options (Figure 2) including Directions (user help feature to assist the user to get started in interacting with the chatbot), New User (for a first-time user), Continuing User (for a user who has earlier interacted with the chatbot and given the chatbot their name), Teach Me (request by the chatbot for the user to teach it), Type Notes/Input Content (for the user to type in learning notes or input content from e-books), View Notes (for the chatbot to display content notes that has been input by the user into it), View Question-Answer pairs (for the chatbot to display the question-answer pairs it has and which correspond to AIML pattern-template pairs), and a Quit option. The buttons on the lower section are Edit Notes (to enable user to make changes to the notes the chatbot has), Edit p-t pairs (for question-answer pair editing), View Currently Unanswered Questions (to enable the user see the questions that the chatbot has been asked during interaction sessions and for which it has no answer i.e. corresponding template), Edit Currently Unanswered Questions Log (make changes to the log of questions that the chatbot does not have answers to). The grayed out buttons (Question, Answer, Keyword, Definition, Submit) become active once the user begins interacting with the chatbot. Figure 3 shows the chatbot's interface during an interaction session. The user is a New User, the user types in their part of the interaction for the bot to accept inside the YOU text field, and the chatbot displays the interaction session exchange (what the user says and what it says in response) on the upper and lower display areas.

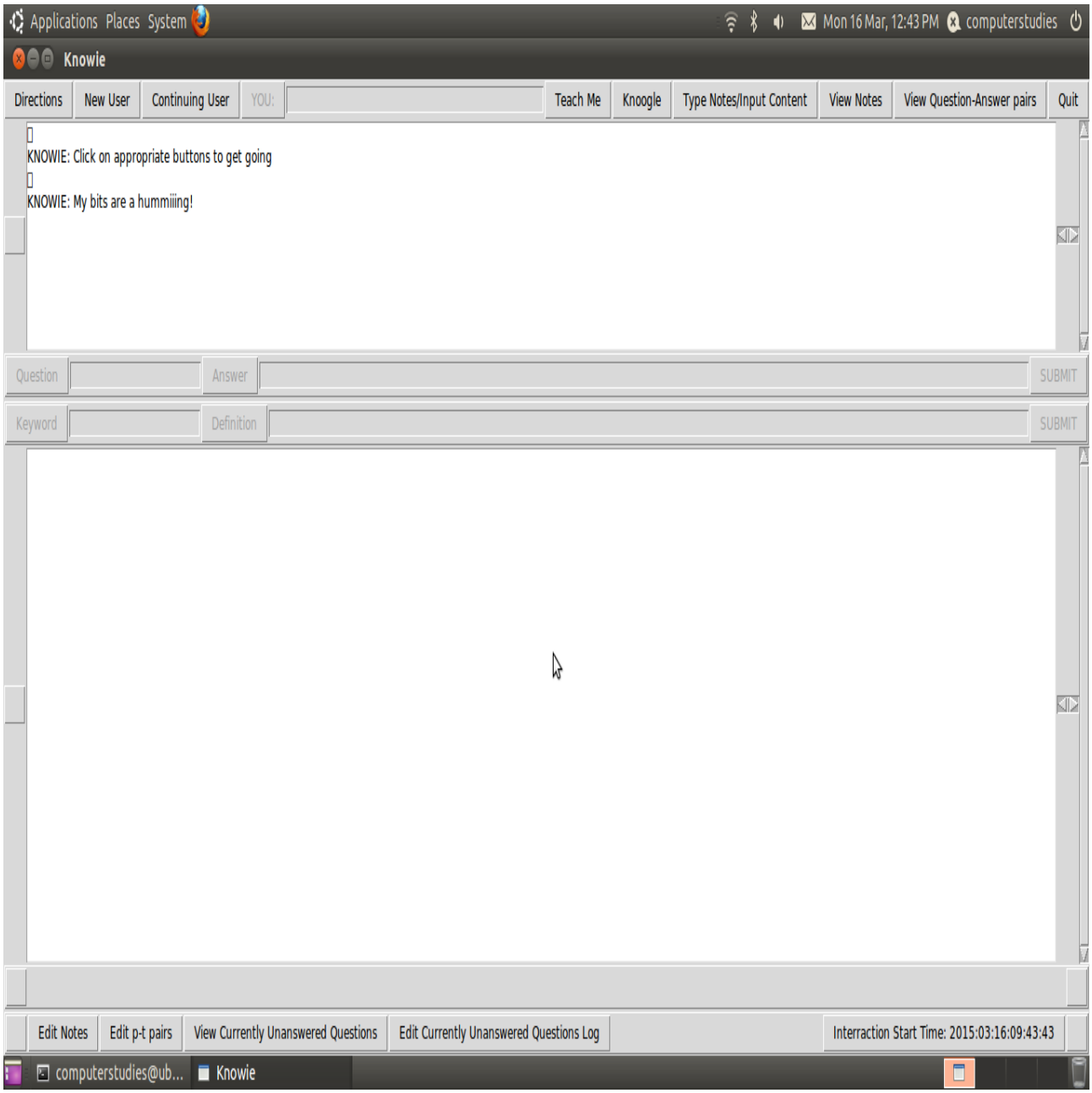


Figure 2. Knowie Basic User Interface

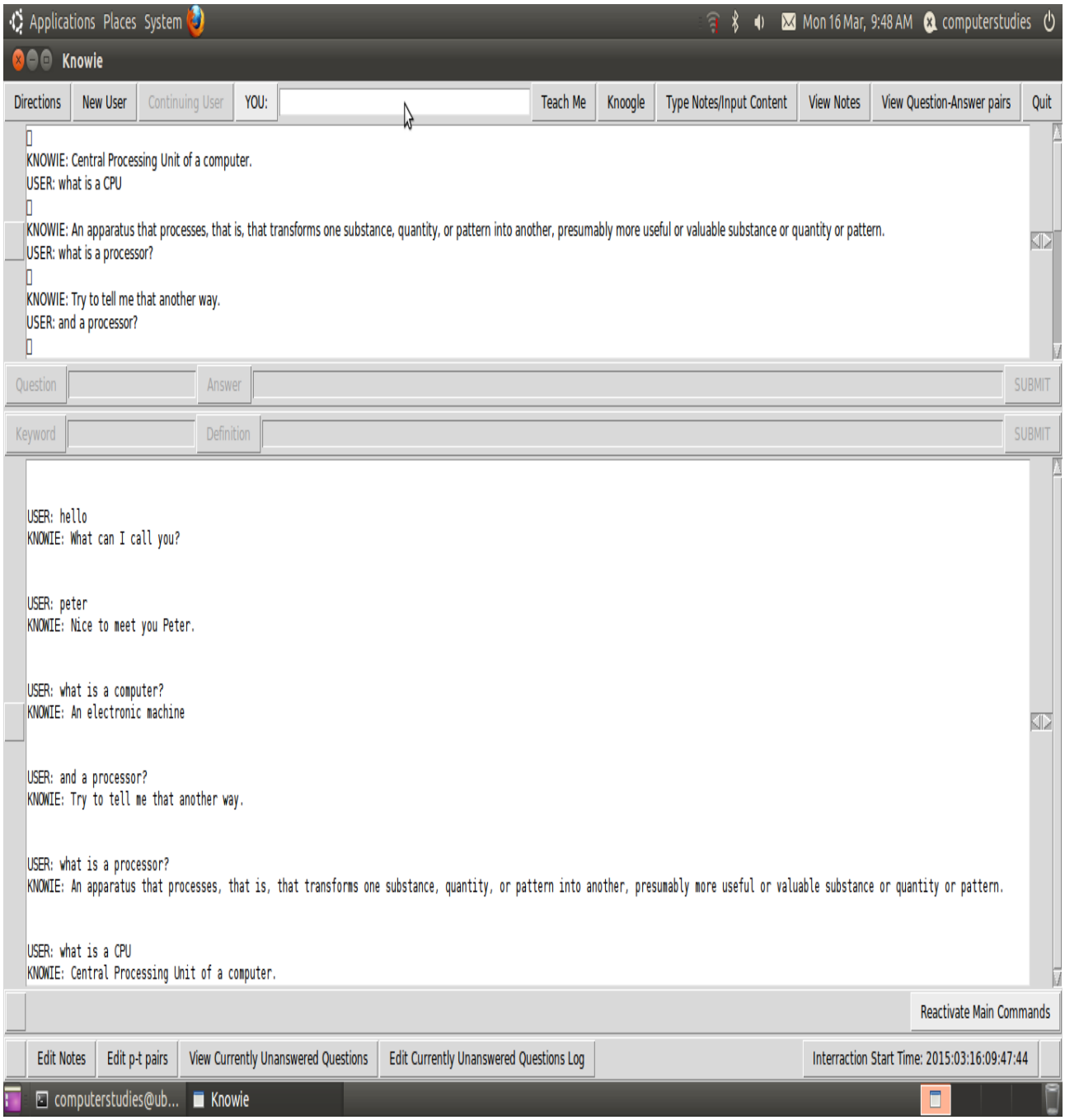


Figure 3. Knowie Sample Interaction Screen Shot

If the user selects Teach Me, the Question, Answer, Keyword, Definition and Submit buttons become active. The user can then program the chatbot by typing in a question (pattern) and the corresponding answer (template) and then submit them to the chatbot to incorporate as part of its knowledge base. The Keyword and Definition buttons serve the same purpose, but in this case, the user types in a keyword (for example Computer) and the appropriate definition of the keyword. Upon submission of the Keyword-Definition pair, the chatbot will proceed to form pattern-template pair combinations from the two using as many as possible ways of asking questions about the submitted keyword, enabling it to generate an answer based on the definition of the keyword and whose answer is the definition supplied by the user. The search (Knoogle) button allows the user to type in a keyword or phrase and submit to the chatbot to retrieve and display any content material that it has about the search term. If the keyword being searched for is found to have a definition, then the appropriate pattern-template pairs for it are also generated, enabling the chatbot to latter respond to a question asked about the keyword by the user.

The chatbot's underlying structure and architecture is illustrated in Figure 4. The GUI Chat Interface registers user

input and displays user response. The user interface acts as the front-end, allowing the student to communicate with the chatbot. It also enables the student to input notes, save notes, view notes, edit notes, search notes, input question (pattern) – answer (template) pairs, view and edit pattern-template pairs, input keyword -definition pairs, and edit keyword-definition pairs. The GUI also displays the questions that the chatbot currently does not have an answer to as an aid for further targeted response improvement. The AIML Interpreter performs user text input processing and generates the bot's response. The interpreter acts as a link between the user and the knowledge base of the bot. It takes the user input, passes it through a normalization process, looks for matching patterns in the database, extracts a suitable response from the database and displays it to the user. The AIML Knowledge Base acts as the back-end which stores the bot's knowledge. Knowledge Base Sources are the various ways in which one can add knowledge into the chatbot's knowledge base and the various online resources the chatbot can access in order to add to its knowledge.

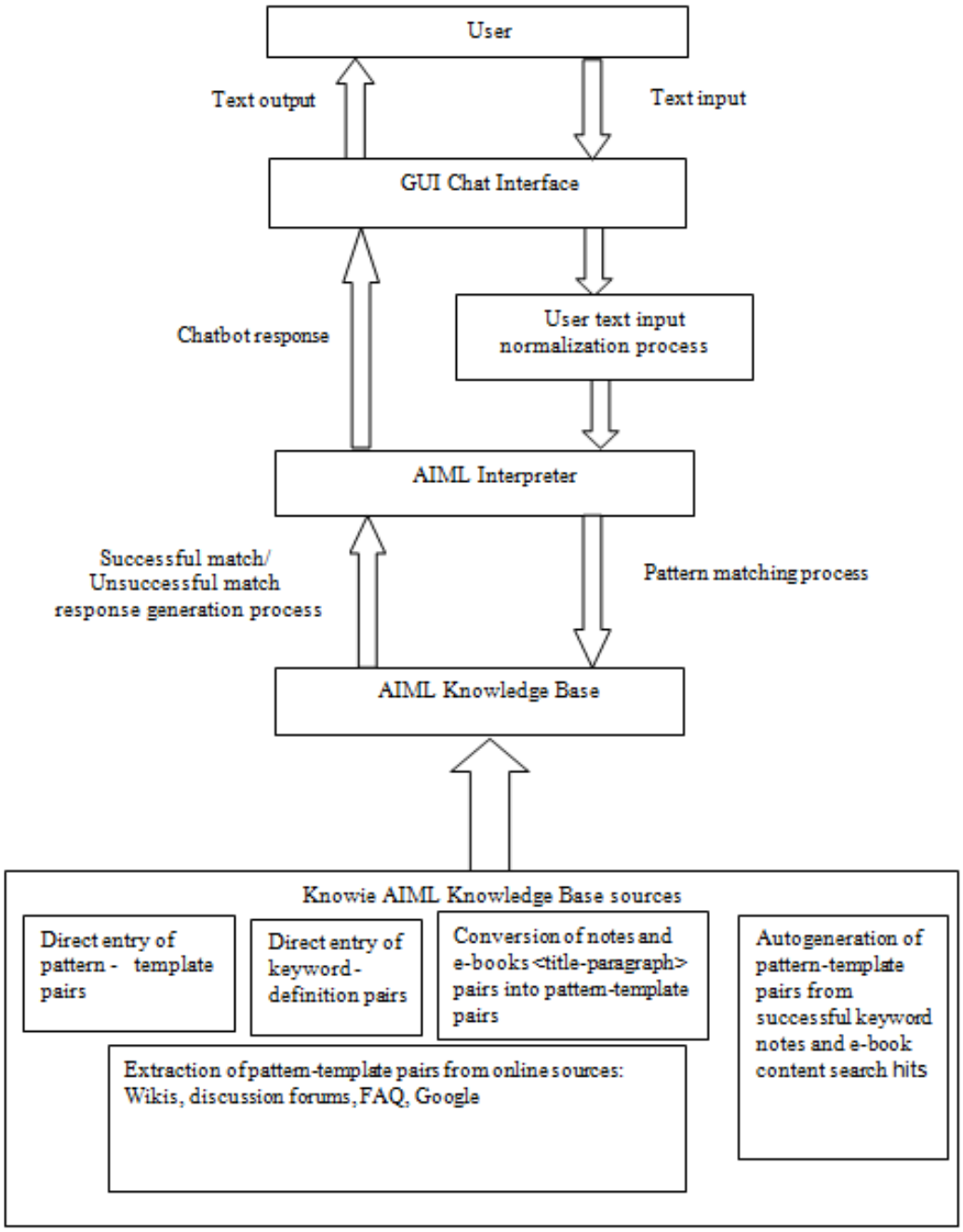


Figure 4. Chatbot Knowie's underlying structure and architecture

The foregoing discussion indicates that it is possible to transfer chatbot technology use to students, teachers, resource persons and other interested developers because of the inherent simplicity involved in AIML design and use. This means that the power of Artificial Intelligence is within reach of the normal classroom environment and such use by teachers and students can be a way to begin involving the key players in the teaching and learning environment - students and teachers - in active development of the content they use in schools. Searches similar to that offered by internet search services is possible, and with further development efforts, can extend to live comprehensive online internet searches.

6. Conclusions

Once computer hardware is availed to schools as PC's, Laptops and other mobile computing devices, a critical issue which must adequately be addressed is the educational software content within. This paper has outlined a possible approach using open source software development concepts and the PAR model as a means of empowering students and teachers to become active developers of the content they consume.

REFERENCES

[1] IST Africa Consortium. *Current ICT Initiatives and Projects-Republic of Kenya*. Available from: http://www.ist-africa.org /home/default.asp?page=doc-by-id&docid=5181 [Accessed 18th June 2013]

[2] Gitonga, A. Enhancing Information Literacy for Vision 2030 and Beyond: Introducing quality to education. Kenya Studies Review Volume 1 Number 2 December 2010.:37-56. Available from: http://kessa.org/yahoo_site_admin/assets/docs/3_ A_Gitonga.140120221.pdf. [Accessed 22nd December 2011]

[3] Government of the Republic of Kenya. *Kenya Vision 2030: The Popular Version.* Available from: http://www.google.co.ke. [Accessed 22nd December 2011]

[4] Kinyamu, M. *Free Laptops for Pupils in Kenya: A Guide on Implementing the Project.* Available from:http://www.slideshare.net/muthurikinyamu/free-laptops-for-pupils-in-kenya-a-guide-on-implementing-the-project [Accessed 18th June 2013]

[5] Jubilee Coalition Manifesto. Shared Manifesto of the Coalition between The National Alliance (TNA) The United Republican Party (URP), The National Rainbow Coalition(NARC) and The Republican Congress Party (RC). Available from: http://softkenya.com/jubilee-coalition-manifesto/ [Accessed 18th June 2013]

[6] Wokabi, C. *From Lollipop to Laptop*. Daily Nation [Internet]. 2013 Apr 16 [cited 2013 Apr 17]. Available from: http://www.cskonline.org/about-us/kenya-ict-press/240-from-lollipop-to-laptop

[7] Katlic, T. *Kenyan student laptop program moves on amid criticism*. Available from: http://www.oafrica.com/education /kenyan-student- laptop-program-moves-on-amid-criticism/ [Accessed 24th July 2013]

[8] Anne, S. *PC's For Children: Can the idea work in Kenya?* Available from: http://www.ihub.co.ke/blog/2013/05/pcs-for-children-can-the-idea-work-in-kenya/ [Accessed 14th May 2013]

[9] Mungai, M. *12 Challenges facing Computer Education in Kenyan Schools*. Available from: http://www.ictworks.org/2011/09/12/12-challenges-facing-computer-education-kenyan-schools/ [Accessed 14/4/2013]

[10] Hepburn, G. Open Source Software and Schools: New Opportunities and Directions. *Canadian Journal of Learning and Technology* 2005; 31(1). Available from: http://cjlt.csj.ualberta.ca/index.php/cjlt/article/view/150/143 [Accessed 24th July 2013]

[11] Hepburn, G. Seeking an educational commons: The promise of open source development models. *First Monday* 2004; 9(8) Available from: http://firstmonday.org/issues/issue9_8/hepburn/index.html. [Accessed 24th July 2013]

[12] Mathieson, J. *Linux Overview*. IEEE Monthly Meeting. 3rd February 2009. Available from: http://ewh.ieee.org/r3/central_georgia/ieee_linux.ppt [Accessed 6th August 2013]

[13] Free Software Foundation. GPLv3. *GNU General Public License*. 2007. Available from: http://www.gnu.org/licenses/gpl-3.0.en.html [Accessed 6th August 2013]

[14] Hepburn, G., Buley, J. Getting Open Source Software into Schools: Strategies and Challenges. Innovate: Journal of Online Education. 2006; 3(1). Available from: http://nsuworks.nova.edu/cgi/viewcontent.cgi?article=1114&context=innovate [Accessed 24th July 2013]

[15] DJX. *A Brief History of LibreOffice.* Available from: http://ubuntero.info/apps/a-brief-history-of-libreoffice/ [Accessed 6th August 2012]

[16] Kroah-Hartman, G., Corbet, J. & Amanda, Mc. Linux Kernel Development: How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It: An August 2009 Update. Available from: http://www.linuxfoundation.org/sites/main/files/publications/whowriteslinux.pdf [Accessed 6th August 2013]

[17] Wallace, R. S. *The Anatomy of A.L.I.C.E.* In: R. Epstein, et. al. Parsing the Turing Test. Springer Science+Business Media B.V. 2009. p.181-210.

[18] Astleitner T, Barnes R, Belzunce A, Carrera, D, Detwiler R, Harpe SE. et.al *Getting Started with OpenOffice.org 3*. 2008. Available from: http://www.openoffice.org/ documentation/manuals/userguide3/0100GS3-GettingStartedOOo3.pdf [Accessed 6th August 2013]

[19] Green, LW, George MA, Daniel M, Frankish JC, Herbert CP, William R. et. al. in Meredith M and Wallerstein N (eds) Community-Based Participatory Research for Health: From Process to Outcomes. San Francisco, CA: Jossey-Bass Inc. 2003.p.419

[20] Milligan, C. (2013). *Participatory Research: Background and Outlook*. Available from: http://www.lancs.ac.uk/researchethics/5-2-outlook.html. [Accessed 8th February 2013]

[21] McTaggart, R. *Participatory Action Research: issues in theory and practice*. Educational Action Research 1994 2(3). p.313-337. Available from: http://www.tandfonline.com/doi/pdf/10.1080/0965079940020302. [Accessed 8th February 2013]

[22] Quixley, S. *Participatory Action Research: An Outline of the Concept*. Available from: http://www.suziqconsulting.com.au/free_articles_files/CD%20- %20PAR%20Detailed %20Overview%20-%20Aug08.pdf. [Accessed 8th January 2013]

[23] Dryden, G. *MakeAiml: An AIML creation tool*. Available from: http://makeaiml.aihub. org/tutorials/aiml_template.php. [Accessed 10th October 2009]

[24] Wang, Y. Designing Chatbot Interfaces for Language Learning: Ethnographic Research into Affect and User's Experiences [Dissertation]. Vancouver. The University of British Columbia, 2008

[25] Abu Shawar, B Atwell, B. Chatbots: Are They Really Useful? *LDV-Forum* 2007 - 22 Band (1), 29-49.

[26] Kerly A, Hall P, Bull S. *Bringing Chatbots into Education: Towards Natural Language Negotiation of Open Learner Models*. 2007. 20(2). p.177-185. Available from: http://www.eee.bham.ac.uk/bull/papers-pdf/AI06.pdf [Accessed 20th June 2011]

[27] Batista AF, Maria GB, Gilseno CO, Barbosa RS, EmersonAN. Multi-Agent Systems in a Computational Environment of Education: A Chatterbot Case Study. *International Journal of Infonomics (IJI)*. 2010. 3(3). Available from: http://www.infonomics-society.org/IJI [Accessed 22nd November 2011]

[28] Knill O, Carlsson J, Chi A, Lezama M. *An Artificial Intelligence Experiment in College Math Education. Havard Mathematics Department.* [preprint]. 2004. Available from: http://www.math.harvard.edu/~knill/preprints/sofia.pdf [Accessed 24th July 2013]

[29] Voog J, Dede C. *TWG 6: 21st century learning*. Available from: http://www.edusummit.nl/fileadmin/contentelementen/kennisnet/EDUSummIT/Documenten/2011/6_EDUsummIT_2011_21st_century_learning.pdf [Accessed 6th August 2013]

[30] Pacific Policy Research Center. 2010. *21st Century Skills for Students and Teachers.* Honolulu: Kamehameha Schools, Research & Evaluation Division. Available from: http://www.ksbe.edu/spi/PDFS/21%20century%20skills%20full.pdf [Accessed 6th August 2013]

[31] Burkhardt G, Monsour M, Valdez G, Gunn C, Dawson, M. *enGauge® 21st Century Skills: Literacy in the Digital Age.* Available from: http://pict.sdsu.edu/engauge21st.pdf [Accessed 6th August 2013]

[32] Stratton C. *PyAIML (a.k.a. Program Y): A Python AIML Interpreter.* Available from: http://pyaiml.sourceforge.net/ [Accessed 12th November 2009]

[33] Esplana PA. *Basic Tutorial On Artificial Intelligence Markup Language (Aiml*) Via Notepad. Available from: http://www.instructables.com/id/basic-tutorial-on-artificial-intelligence-markup-l [Accessed 28th October 1999]

[34] Wallace RS. *AIML Overview*. Available from: http://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html [Accessed 18th March 2010]