International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319-6734, ISSN (Print): 2319-6726
www.ijesi.org ||Volume 9 Issue 1 Series. III || Jan.2020 || PP 31-37

An Enhanced Knowledge Engineering Framework for Knowledge Acquisition and Validation

Odulaja Godwin Oluseyi[1], Awodele Oludele[2], Idowu, Sunday[3], Joshua Vincent[4]
[1,2,3,4]*Computer Science Department, Babcock University, Nigeria*
Corresponding Author: Odulaja Godwin Oluseyi

Abstract: *Knowledge Engineering (KE), a subfield of Artificial Intelligence (AI) focuses on design and production of intelligent Knowledge Based Systems (KBS) such as experts systems. They simulate human cognitive skills and are very useful in many crucial and life-critical specialist fields. Development of KBS is contingent on availability of task domain knowledge providers for knowledge elicitation. Paucity and lack of cooperation of domain experts had led to poor design and performance in KE. This makes validating performance of KE products difficult. These situations can improve given a KE framework that factor addressing knowledge elicitation challenges into design of every KE product. Previous studies had focused mainly on formation, refinement, structuring and reusability of knowledge models with particular emphasis on production of standard electronic knowledge acquisition and documentation models and templates. This study developed an improved KE framework that addresses knowledge elicitation challenges. The framework - Integrated Expert System Design and Development Approach (IESDDA) develops KE products in six phases and uses an Experts Consultative Interactive Submodule (ECIS) for consulting, convincing, encouraging and collaborating with knowledge domain experts in order to gain their trust and support for populating the products' knowledge base, validating products performance and for regular update on a continuous basis.*
Keywords*: Artificial Intelligence, ECIS, Expert Systems, IESDDA, Knowledge Engineering.*

Date of Submission: 28-01-2020 Date of Acceptance: 09-02-2020

I. Introduction

Expert Systems (ES) are the major forms of Knowledge Based Systems (KBS) produced from Knowledge Engineering (KE) concepts under Artificial Intelligence. They are applications that are specially designed and developed to carry out the role of a human expert as a substitute or in the least, to support the human expert in exhibiting his cognitive prowess [1]. KBS are highly needed products in everyday life and are found in crucial and life-critical areas such as food production and life rescue missions.

Major challenges facing KE products are in connection with design and performance. A significant cause of performance and design issues in KE products is the paucity of knowledge domain experts that are willing to cooperate with knowledge engineers and developers. This problem results in corresponding difficulties associated with knowledge elicitation (acquisition), validation by domain experts [2]; [3] and product integrity issues [12]. These problems had also resulted in KE products that failed to meet changing user requirements specifications as new technologies (such as mobile and ubiquitous computing) emerge.

II. Statement Of The Problem

Knowledge Based Systems (KBS products are now besieged with many design and performance related challenges resulting from difficulties associated with knowledge elicitation from domain experts. Knowledge Engineering practice requires integration of domain specific knowledge into the computer systems [5]. When Knowledge Engineers could not gain the support of domain experts and elicit requisite knowledge from them, validating any product consequent from this becomes practically impossible. This undermines performance, reliability and durability of such products regardless of how well refined the model used can be [1].

It became apparent that both design and performance of KE products are due for thorough methodology overhauling in order to meet modern day user requirements and reflect state-of-the-art computing.

Corrective efforts in literature had focused majorly on formation, refinement, structuring and reusability of knowledge models as found in CommonKADS and MIKE with particular emphasis on production and refinement of standard electronic knowledge acquisition and documentation models and templates [6]. These measures however did not address the problem of knowledge elicitation because only available knowledge can be captured onto the existing model templates. Besides, knowledge that has not been acquired cannot be modified or refined. In situations where domain experts are not cooperating in giving needed knowledge, such knowledge capture models and templates become irrelevant. Even if the knowledge is sourced elsewhere, perhaps through existing literature, validation of the system built on such will be unreliable without involvement of willing domain experts.

Therefore, this study proposed a knowledge engineering framework – Integrated Expert System Design and Development Approach (IESDDA) for KE products development and practice (KEPP) that addresses the problem of knowledge elicitation in KE in general, and especially to eliminate the knowledge acquisition, validation and update challenges.

III. Related Works

Atanasova and Krupka [7] in their Modern Expert System Architecture reflected the presence of new elements such as interactive module, coordination module, rule editor and knowledge editor in design of modern ES. They also separated knowledge acquisition from ES shell. Saket, Akankasha and Vikas [8] explored the structure and role of expert system in agriculture and gave an overview of feasibilities of designing, developing and implementation of expert systems for agriculture. Their objective is to motivate scientists and extension workers to peer into possibilities of developing applications for the benefit of entire global agricultural community. The work however lacks specific implementation framework that factors emerging technologies into ES design. Diana-Aderina [9] specified that expert system components have gone beyond consisting of just the traditional three components: inference engine, user interface and knowledge base and there is the need to acknowledge this in design of modern ES.

Wiliyanto,[10] in a study conducted to determine the effectiveness of web based expert system applications, found expert systems effective in identifying and intervening to help children with special needs in inclusive schools. She recorded 52% result accuracy. Peter Lucas [11] of the Institute for Computing and Information Sciences, Radbound University Nijmegen, Netherlands, concisely defined modern Expert System in terms of just two major components namely: **Knowledge** and **Problem Solving Methods.** He admitted that the architecture of an expert system can only be defined in terms of constituent components of the ES as well as in terms of the interchange of information between these components. This implies that modern ES may contain other components such as explanation facility and trace facility (Fig.2).

Shodganga [12] compared several ES development models applicable in Agric and came up with a web based model (Fig. 1). He pointed out that ES is an expertise driven research area where researchers have many times used multi methodologies approach to develop ES. He added that Rule based system and KBS are the two methodologies often used for Agricultural ES. A flaw in the model however is the inability of the system to autonomously validate input and prevent data redundancy. The model also does not consider online knowledge acquisition and validation against security vulnerabilities such as when a non-domain expert poses as a domain expert. Thus, there is the need to develop a framework that autonomously verify and validate submitted piece of data or knowledge for validity and to avoid data redundancy.

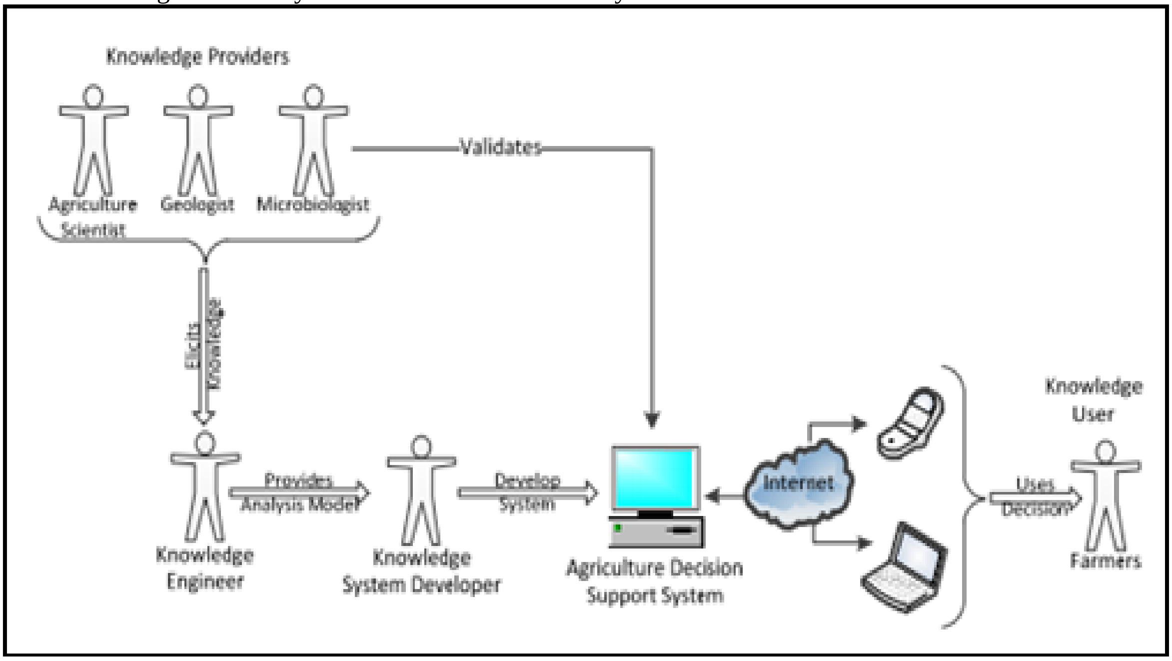

Figure 1: Organisation model of a web based agricultural decision support system
Source: Shodhganga, 2018

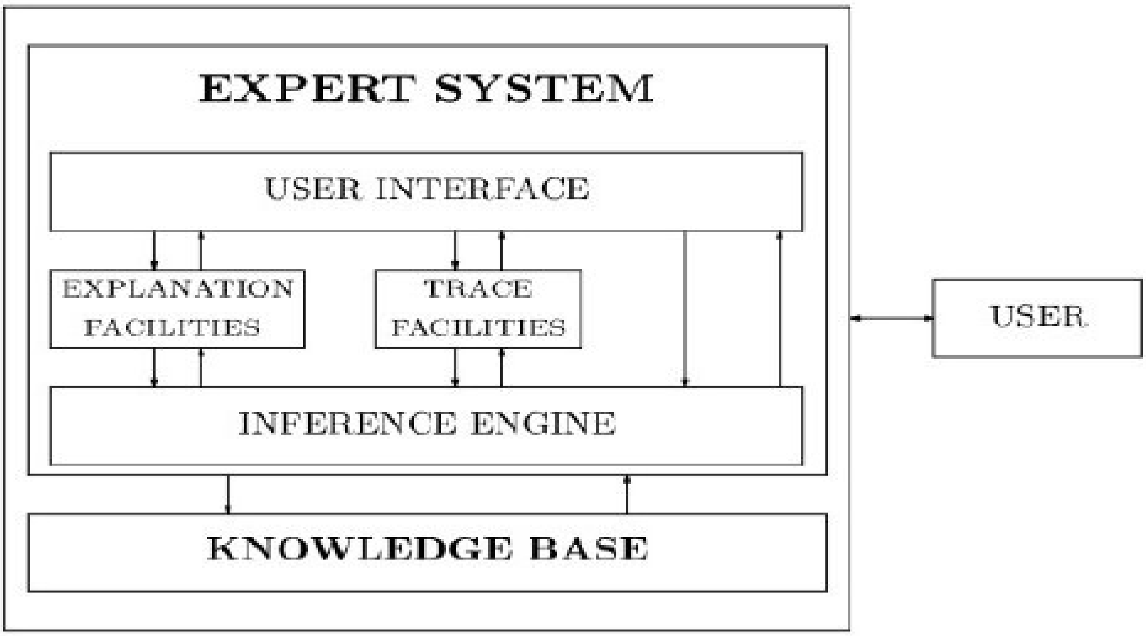

Figure 2: Global expert system architecture

Source: Lucas, 2017

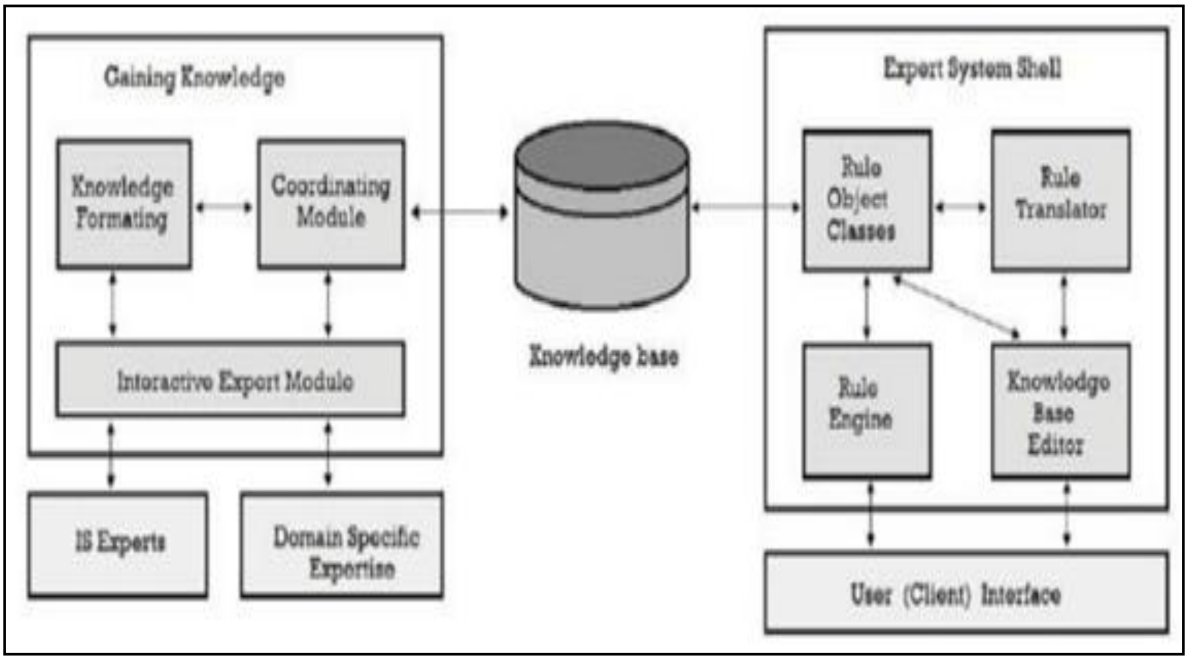

Figure 3: Modern expert system architecture.

Source: Atanasova & Krupka, 2013

In their Modern Expert System Architecture (Fig.3), Atanasova and Krupka [7] reflected the presence of new elements such as interactive module, coordination module, rule editor and knowledge editor in design of modern ES. Nevertheless, this design however did not factor online autonomous knowledge acquisition, update and validation into their design model. It also left out referral feature characteristic of human experts at all levels of expertise. Shodhganga's[12] shows the need for domain expert involvement during knowledge acquisition stage. It however left out performance assessment metrics, feedback mechanisms, and referral features. It is apparent that none of these works practically and fully addresses the problem of knowledge elicitation.

IV. Methodology

IESDDA is an enhanced SDLC (Software Development Life Cycle) model. It is a software development model created for developing Expert Systems or Knowledge based Systems. It is a synergy of the strengths of Spiral model, Rapid Application Development (RAD) model, Rule based expert system (RBES) development and Knowledge Acquisition and Documentation Structuring KADS.

Like RAD, IESDDA favors a speedy iterative approach through regular modular testing and integration, and as in Spiral Software Development approach, it incorporates early risk detection, control and management through consistent user involvement right from problem conception and identification stage all through to design, development and testing stages. As found in KADS, standard software development models are used to simplify complexities in knowledge engineering processes thereby making careful conversion of expertise models into realizable system design for implementation realizable. During implementation, KE products of IESDDA can be designed to adopt Role Based Access Control (RBAC) to assign control or restrict access rights for role players. The six developmental stages of IESDDA are given and discussed in section 3.4.

4.1 Developmental Phases of Integrated Expert System Design and Development Approach (IESDDA)

The six development phases of IESDDA are here discussed in sequence, one after the other.

4.1.1 System Requirement Elicitation, Acquisition and Analysis

This first stage – System Requirement Elicitation, Acquisition and Analysis has three substages:

a. Initiation – Problem definition and description is followed by Elicitation and Acquisition of System Requirements
b. System Requirement Analysis and Concept Development
c. Output from (a): User Requirement Specification

At the **Initiation** stage, IESDDA like SDLC requires that the problem to be solved be clearly defined and described in enough detail to clarify what knowledge or data must be acquired from the users and domain experts, and what are the expected system requirements and outputs. At this stage the user departments of domain experts are contacted and their expectations in terms of system requirements are acquired by the knowledge engineer and documented. This stage is very crucial in the developmental life cycle of the ES in that the Knowledge Engineer must skillfully and persuasively establish concrete and ongoing mutual relationship with the domain experts whose support, input and cooperation will be needed throughout the life of the software. He should at this stage introduce and establish communication and contact procedures that are mutually acceptable and convenient especially to the domain experts. These could be online, by telephone or by physical presence as the case may be. This is also an early stage where the proposed system could be terminated if it becomes apparent that it is not feasible. The proposed development may not be feasible for instance if domain experts are not willing to cooperate with or support the project or other sources of needed knowledge and funding could not be guaranteed. So IESDDA creates an **exit** gate here.

The second substage - System Requirement Analysis and Concept Development is the stage where obtained system requirements from substage 1 are analyzed. A feasible corresponding conceptual framework is built based on input from substage 1.

The third substage generates user requirement specification as a direct output from refining inputs from substages 1 and 2. The members of the management or their representatives obtain a copy of this document for analysis, modifications, ratification, or outright rejection if deemed unattainable. So IESDDA creates another **exit** gate here. Otherwise, the next stage ensues.

4.1.2 Preliminary System Design, Specification and Formalization

This is the second stage of IESDDA methodology. Output from this stage is a formalized System Requirement Specification (SRS) prepared with the support of system analyst(s). The analyst breaks down the project into modules and submodules and formalizes it into appropriate integration-specific modular structure for coding by the team of programmers. This serves as the template or blueprint for building or coding the proposed system. Before embarking on the next stage, the stakeholders are served copies of this blueprint for perusal. If need be, corrections for necessary adjustments are received and implemented before coding begins. If any fatal error is realized, **exit** gate is taken.

4.1.3 Building/Coding, Testing and Refining the first prototype with domain experts.

At this stage, the knowledge engineer and the system analysts engage the programmer(s) for coding (using appropriate programming language that suits the problem most) to build the first prototype. Considering performance of the first prototype based on the content of the Formalized System Specification generated in the previous stage, the services of domain experts are also solicited at this stage for testing and subsequent refinement of the first prototype. Risks, errors, conditions and constraints are identified, fixed early or tested as the case may be. Corrections made on the first prototype metamorphosed it into the second prototype.

4.1.4 Iterative Building/Coding, Testing and Refining of the Second Prototype

There are two substages of this stage: 1) reconstruction of the first prototype and 2) refinement of user requirement specification. At the first sub-stage, features of the first prototype is improved as indicated by the evaluation and assessment team comprising of knowledge engineer(s), domain experts, system analysts and programmers. Test data are run and case studies are explored. The objective is to ensure that the product meets the requirements specified in the user requirement specification.

Sometimes, it may be realistic to adjust, redefine or refine the user requirement specification as necessary. This however must be a collaborative activity.

Adopting evolutionary spiral methodology here enables IESDDA to overcome a peculiar gap or weakness of the otherwise disciplined approach of traditional SDLC – putting too much emphasis on planning which necessitates that all details be clarified ahead of design and implementation. This lack of flexibility is a weakness that neither leaves room for inevitable error nor provide avenue for handling feedback on inevitable problems surfacing during development. This weakness of SDLC is taken care of as IESDDA allow for spiraling, feedback and feed-forward during design and development of application.

Another problem is the interstage transition decision problem. That is before transiting to each new stage, SDLC does not allow room for management decision on whether to move on or terminate the process, especially when it becomes apparent that the project will not sail through profitably. However, at the end of iteration, IESDDA introduces a gate for development team and management to decide whether to continue development based upon the potential for profitable implementation or not.

Although this stage is iterative, there must be a termination point or condition - a set limit or degree of user satisfaction when the first version of the software can be released. Otherwise the iteration will experience scope creep, or enter an endless loop.

Together with the user department or domain experts, knowledge engineers, analysts and programmers hold a stakeholders meeting to determine what degree of satisfaction or performance could become the exit or termination point to stop the iteration and release the software. This is documented for implementation at the appropriate time.

4.1.5 Integration, Documentation and Evaluation – System (Software) Reliability Test

At a point and as indicated in stage iv, the software attains the stipulated exit or termination level of satisfaction for release. At this point, the software is tested and retested by the stakeholders including the domain experts, knowledge engineers, programmers and analysts for functional reliability with the requirement specification as the yardstick. Integrating the KE product could be achieved in two ways. One is the combination of related sub-modules of the software into functionally coherent module and further combining the modules together to form the expected software. Integration is otherwise called installation – integrating the new software into the system of the organization(s) that will use it. This too might take place in two ways: offline and online. It is offline for proprietary systems and for virtual private networks who sponsored the development of the application for local and private use. On the other hand, the software might be deployed, hosted and uploaded onto the cloud or be made web-based for wider global access. Reliability and security checks are crucial at this stage.

4.1.6 Operation, Maintenance and Adaptive Scaling

This last stage involves the user department, domain experts and the administrator or knowledge engineer. The installed and integrated software goes into use. Users are trained on use or if users are computer and web-literate already, they are allowed to practice "Do-it-Yourself. (DiY)" and become familiar with features and operations of the system. A seamless changeover is carried out. IESDDA recommends **pilot** changeover which favors using a particular department or group of individuals or related organizational units as forerunners in using the software. Obtained usage result from pilot user departments helps management determine whether to extend the use of the software to other departments or not and how soon. If the decision is in favor of adoption of the software, together with the knowledge engineers and the developers (analysts and programmers), the system administrator (or a team of **Content Developers)** is saddled with the responsibility to maintain, secure, update, upgrade and manage the software functionally and content-wise.

As new discoveries (in the knowledge domain) and applicable technologies emerge, the system administrator monitors the system as it autonomously selects, validates and absorbs new knowledge autonomously. The team also makes upgrades available to users automatically on regular bases. As the number of individual and corporate users increase, the team recognizes the need for **adaptive scalability**. That is, a controlled expansion of the scope and capability of the software such that more users can be accommodated without performance degradation. It also involves knowledge base content update without unnecessary duplication of input. Figure 3.3 shows the schematic diagram of IESDDA. This is made easier by the modular structure of the software development process endorsed in IESDDA.

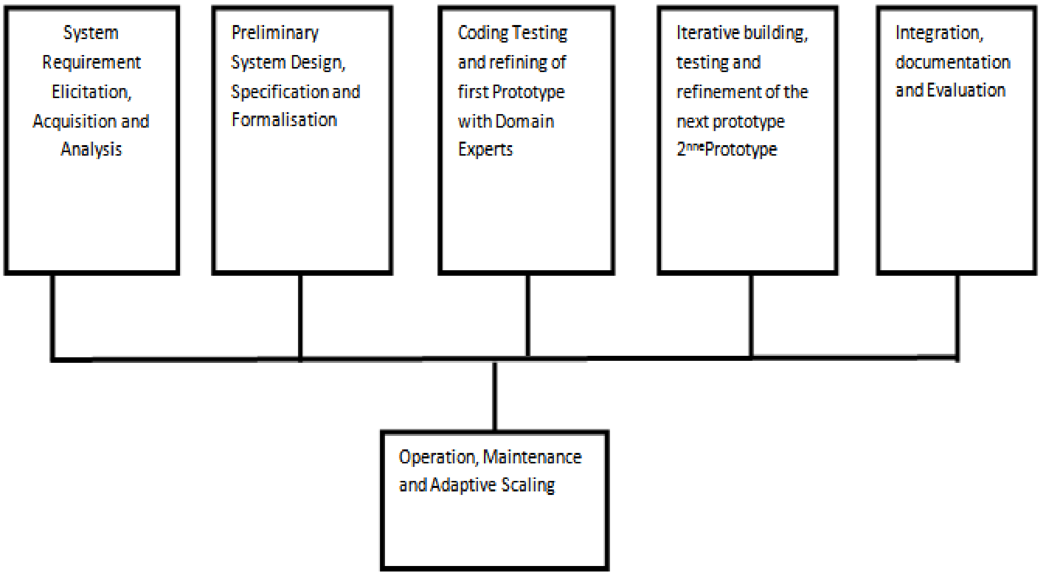

Figure 4: Schematic diagram of IESSDA software development model

V. Architectural Design Of IESDDA

The architectural design of IESDDA model (Figure 4) shows its three major divisions: the **knowledge acquisition module/phase,** the **server side,** and the **client side.** The Experts' Interactive Consultative Submodule of the knowledge acquisition division is the most crucial to the success of the entire project. This is because knowledge elicitation, product performance and functional validation, as well as product reliability are contingent on the success of this stage. Thus this module is specifically designed to address these cogent issues. In this section, domain specific (knowledge) experts, System Developers, and Knowledge Engineer(s) meet. The objective is for Knowledge Engineer to use this forum to persuade, convince, encourage and elicit domain specific knowledge from knowledge domain experts. This phase tests human relation skill of the Knowledge Engineer as it requires tact, persuasiveness, and sometimes motivation in cash or in kind. It may be necessary to enlist the support of skilled human relation officer as research assistant in this regard. This may also require a series of interactive consultation fora. Such generated knowledge is subjected, through the coordinating module, to refinement, formatting and necessary validation before it is transferred into the knowledge base. This considerably takes care of knowledge acquisition challenges facing knowledge engineers in the course of developing KE products.

The server side comprises of the knowledge base (of facts and rules) and the expert system shell. The expert system shell comprises of eight subcomponents interacting together to respond to request from the clients. These eight components are: the rule engine, the rule translator, explanation facility, working memory, rule object classes, knowledge base editor, meta knowledge and intent analysis and pattern matching subcomponent which receives input directly from users. Through the user interface, each user passes his request to the intent analysis and pattern matching subcomponent of the ES shell in form of https request. This request is analyzed to determine the intention of the user. Through pattern matching algorithm, appropriate and corresponding response will be generated and returned to the user as https response.

The client's side comprises of the users passing their queries in form of https requests through the user interface systems and web application services to the expert system shell. The https response is received via the same channel for users' consumption (Fig.5).

Leveraging on existing models and the identified gaps in literatures, IESDDA design allow for autonomous knowledge capture, validation and update. Through its iterative interactive consultations with experts, it allow for early risk detection and management (Fig.5). Through features like referrals, feedback and feed-forward, and share resource, the model allows for knowledge transfer, sharing and regular update.

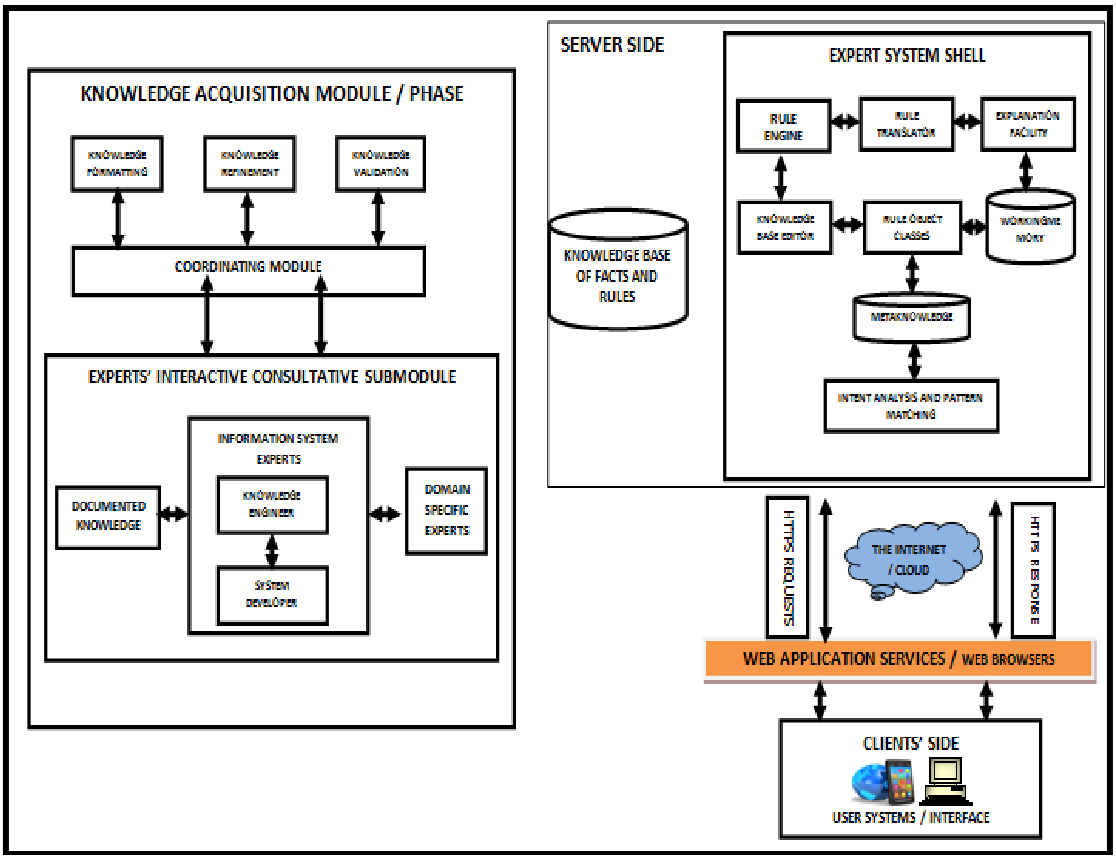

Figure 5: Architectural design of Integrated Expert System Design and Development Approach (IESDDA) framework

VI. Conclusion

This study developed an improved Knowledge Engineering framework that addresses knowledge elicitation (acquisition) and validation problems in KE. Adoption of IESDDA framework in KE will prove helpful because the framework promotes early involvement of domain experts in KE product development. It also encourages online autonomous knowledge acquisition, collaborative KE product development and early risk detection and management. This results in reduced product development down time and consequently increases product durability, performance and relevance. Further research could focus on integrity issues in online autonomous knowledge acquisition.

References

[1]. S. Rudi, V. Benjamins, and F. Dieter. Knowledge Engineering: Principles and Methods, retrieved from https://www.researchgate.net/profile/V_Richard_Benjamins/publication/222305044_Knowledge_engineering_principles_and_methods_Data_Knowl_Eng_251-2161-.pdf 2017

[2]. S. Osindero, S. and Y. Teh, Y. A fast Learning Algorithm for Fast Learning Nets,Neural Computations,18(7), 2006, 1527–1554.

[3]. S. Ele, E. Umoh, and W. Adesola, An Overview of the Development, Principles, stages and Building Blocks of Expert System, West African Journal of Industrial & Academic Research, 2(1) 2014

[4]. Shodhganga (2018). Agricultural Expert System, retrieved from http:// shodhganga.inflibnet.ac.in/bitstream/10603/71534/10/10_chapter%203.pdf.

[5]. G. Falquet, Knowledge Engineering and Representations http://cui.unige.ch/isi/icle-wiki/_media/cours:mttsd:knowledge_engineering_and_representation.pdf

[6]. P. Sureephong, N. Chakpitak, Y. Ouzroute, G. Neubert, and A. Bouras Knowledge Engineering Techniques for Cluster Development. retrieved fromhttps://www.researchgate.net › publication › 301872111_Knowledge_Engin... 2016

[7]. I. Atanasova,, and J. Krupka. Architecture and Design of Expert System for Quality of LifeEvaluation, Informatica Economică , 17(3), 2013, 28-35.

[8]. S. Saket and D. Vikas, Expert Systems In Agriculture: An Overview;International Journal of Science Technology & Engineering. 1(5), 2014.

[9]. M. Diana-Aderina & A. Mihai-Constantin, Architectural Model of Expert Systems, International Symposium Engineering and Competitiveness (EMC 2015) https://www.researchgate.net/publication/

[10]. D. A. Wiliyanto, The use of web based expert system application for identification and intervention ofchildren with special needs in inclusive schools.journal.uad.ac.id 11(4) 2017

[11]. P. Lucas, Artificial Intelligence - Expert Systems,End of Life Support Systems (EOLSS)

[12]. M.Shodhganga,AgriculturalExpertSystem,retrievedfromhttp://shodhganga.inflibnet.ac.in/bitstream/10603/71534/10/10_chapter%203.pdf. 2018