An Analysis of Six Standard Ontology Editing Tools for Capturing Entire Crop Processing

Emmanuel Ukpe & S. M. F. D Syed Mustapha

School of Information and Communication Technology
Asia e University
Jalan Sultan Sulaiman, Kuala Lumpur, Malaysia

Abstract--**In the past decade, the ontology development community has witnessed several platforms and tools for ontology building. All these tools facilitate ontology development processes and direction for the subsequent usage. However, research has shown that current ontology editors do not effectively capture agricultural processes. Existing ontology editors do offer explicit but incomplete agricultural process information. This research proposes the need for a new ontology editor for process capturing, specifically capable of capturing entire cassava plantation process, which can be used to develop Intelligent Tutoring System (ITS) for farmers on crop processing. To this end, this paper examines, analyzes and presents the results of selected ontology editors. The comparison was done using different criteria including an ontology editor's strength, weakness and suitability for capturing entire crop plantation process.**

Keywords: Ontology editors, Ontology, Protégé, Apollo, KAON2, SWOOP, WebOnto & Ontolingua

1. INTRODUCTION

The knowledge base that formalizes all aspects of a particular crop, in this case, cassava, will contribute to the preservation and dissemination of cassava information to aid agriculture professionals in plantation and development of crop process ontology. The crop process ontology is anticipated to be broad enough for adaptation and reusability for other crops in the agriculture domain.

Information on crop plantation process has become critical especially when it comes to issues of soil preparation, crop diseases management, quality and quantity of output to name a few. Thus, the capturing and documenting crop processes have taken center stage in agriculture domain. Knowledge of a concrete and or entire crop process is necessary for the development of Intelligent Tutoring System (ITS) to aid farmers and other agriculture-related professionals. Development of a comprehensive ITS for a particular crop requires information on entire plantation process for that crop. This need for a comprehensive ITS necessitates the requirement for a new ontology editor capable of such information gathering.

This study intends to offer a concept of an ontology editor, capable of capturing entire agricultural production process for a particular crop. The captured data, would enable the design of Intelligent Tutoring Systems (ITS) that cover all operational stages, from soil preparation, planting, pre, post harvesting and other vital information requisites for a crop, such as cassava production life cycle.

II. CASSAVA FARMING in NIGERIA

Knowledge of Cassava farming is of interest, particularly to Nigerians and also, to the rest of the cassava consuming and producing world. Cassava farming knowledge is usually passed on from generation to generation in the families or from trainers to learners in specialized agricultural institutions and or organizational settings. Information on Cassava processing, for example, is currently dispersed, disorganized, and are in varying stages of the plantation [1]. Thus, the continued absence of ontology editors that would accurately capture and preserve the knowledge of entire crop processing would be devastating in the long run. This researcher believes that farming experiences would be lost as farmers gradually shift to other trades due to urban migration and lack of interest of the young generation if nothing is done to preserve such knowledge.

III ONTOLOGIES

Significant research and progress have been made concerning ontology development ideas and editors. Generically, an ontology goal is to gather and organize specific domain knowledge and provide this information in an acceptable standard. The information, include common conceptualizations of a particular domain and the representations of these concepts [2]. This concept of ontology has indeed encouraged and produced numerous ontology editors. Ontology is perceived as a pillar for different types of knowledge management for information storage, retrieval, and sharing.

Ontology design requires the application of software tools, available in commercial or open source, known as Ontology Editors [3]. Such editorial tools can be used at different stages of design, deployment and maintenance of an ontology development life cycle. This paper analyzes some of the popular ontology editors for entire process capturing capability, role, and necessity for constructing ontology editor to support more expressive control and process capture.

IV. ONTOLOGY EDITORS

Ontology editors are used in designing ontology to facilitate excellent information sharing among system users and or software agents. Currently, a variety of development platform exists for construction of ontologies. These platforms are designed for the building of a new ontology either from beginning or reuse of existing ones, to support import and export of diverse formats, viewing and editing capability, browsing libraries and documentation with integrated inference engines. Also, users are provided the opportunity for inspection, visual manipulation, coding, maintenance and other support [4].

In this analysis, we reviewed six popular, standardized and widely accepted ontology authoring tools for constructing ontology schemas, using or without instance data; namely Protégé, Apollo, KAON2, SWOOP, WebOnto and Ontolingua with a focus on the breath, depth of the weakness and suitability of these tools for capturing entire crop process, such as cassava plantation.

4.1 Protégé

Stanford Medical Informatics developed Protégé. Protégé is a Java-based tool equipped with an extensible plug-in architecture, which enables rapid application development and prototyping. Protégé allows a user to construct domain ontologies, create data entry forms, and collect data for added plug-in functionalities. Also, Protégé enables the definition of classes, relationships, and properties, the hierarchy of classes, variables and value restrictions [5]. It is equipped with OWL API that encompasses the core API, which enables access to OWL ontologies. Diagrams and tables are constructed using graphical widgets; however, the addition of new basic types is difficult. On the contrary, Protégé is designed with visualization packages such which help the user visualize ontologies using diagrams. Importantly, for the ontology community, Protégé is a free open-source tool that can be used to construct various knowledge bases [6].

4.2 Apollo

Developed by the Open University of United Kingdom Knowledge Media Institute, Apollo tool provides the user the opportunity to model ontology with basic primitives. The Apollo model is based on Open Knowledge Base Connectivity (OKBC) protocol. Apollo knowledge base consists of hierarchically organized ontologies, which can be inherited from other ontologies. Inherited ontology usually contains all primitive classes, such as Boolean, float, integer, list, and string to name a few. The class contains template and non-template slots, which can be used to generate instances. Apollo is written in the Java language, not bound to any language and can be extended to different formats of I/O plug-ins. Also, it allows implementation of other knowledge bases, but it does not support collaborative work [7].

4.3 KAON2

KAON2 is a framework for managing OWL-DL, F-Logic, and Semantic Web Rule Language (SWRL) ontologies. Developed by the University of Karlsruhe AIFB Institute in collaboration with University of

Manchester, Information Management Group (IMG), and Information Process Engineering (IPE) at the FZI Research Center in Germany. KAON2 differs from KAON1 which focuses on business applications; it supports scalability, RDFS extension with symmetric, inverse and transitive relations in addition to efficient reasoning with ontologies and meta-modeling using axiom patterns. KAON2 supports ontology languages such as OWL-DL and F-Logic. KAON2 tool is designed with two user-level applications: KAON PORTAL and OiModeler. All other applications and modules are designed for software development. KAON PORTAL enables ontology navigation and search using a Web browser; while OiModeler is the main editor for ontology creation and maintenance [8].

4.4 SWOOP

Developed by MND University of Maryland, Semantic Web Ontology Overview and Perusal (SWOOP) are an open-source, hypermedia inspired Web-based OWL ontology editor, written in Java. Designed with OWL validation, presentation syntax views and enables multiple ontology environments. SWOOP main features include comparing; creating, editing, and merging of ontologies, with the key features of collaborative annotation, SWOOP is a powerful Web ontology editor. However, it cannot capture process, especially entire crop planting and harvesting process, such as cassava. SWOOP is known not to follow a particular method for ontology design; neither does it allow fractional imports of OWL [9].

4.5 WebOnto

Developed by the Open University of England, Knowledge Media Institute to support the design, editing of ontologies, and collaborative browsing. WebOnto was constructed using a Java-based central server and encapsulated in OCML knowledge modeling language. The main characteristics of WebOnto are the automatic instance-editing, forms generation from class definitions, inspection of elements, consistency checking, management of ontologies using graphical user interface; support for collaborative work; receiving and making annotations [10/11].

4.6 Ontolingua

Developed by Stanford University Knowledge Systems Lab, OntoLingua as its popularly known is a tool that supports collaborative editing, browsing, creation and distribution of various ontologies. Also known as Ontolingua Server frame-editor has other systems such as Open Knowledge Base Connectivity (OKBC) Server, Webster, and Ontology merge tool embedded into it. Ontolingua, a form-based Web interface was designed to facilitate the development of ontologies. It features support and enable consensus on common shared ontologies. This editor supports collaborative editing, browsing, distribution and creation of ontologies. Also, it provides users opportunity to access and assemble information from a library of modules and reusable ontologies. The user access level assignment and write-only locking functions provide multiple users' concurrent access to Ontolingua. Ontoingua's ontology collection supports and can be accessed through a browser, and it enables translation of different formats [12].

Table 1: Comparative Analysis of Ontology Editors Reviewed

Features	*PROTÉGÉ*	*APOLLO*	*KAON2*	*SWOOP*	*WebOnto*	*Ontolingua*
Availability of Tool	*Open / Free*	*Open Source*	*Open Source*	*Open Source*	*Open Source*	*Free @ evaluation period*
Software architecture: Extensibility, stand-alone, client/server or web-based	*YES*	*YES*	*YES*	*YES*	*YES*	*YES*
Interoperability: enable import & export from languages, merging, annotation, storage,	*YES*	*YES*	*YES*	*YES*	*YES*	*YES*
Inference engine & Exception Handling	*YES*	*NO*	*NO*	*NO*	*YES*	*YES*
Editor usability [ease of use]	*YES*	*YES*	*YES*	*YES*	*YES*	*YES*
Process capturing & modeling: such as cassava plantation & other crops	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

The table above presents a comparative analysis of selected Ontology editors based on the following criteria:

Availability: access to these ontology editors varies, based on developers, most are open source and free, while others are commercial packages. Editors used for these studies are open source and free, which perhaps explain why they are very popular and common.

Software architecture: A significant aspect of ontology editor analysis is the architecture, which covers platform information, stand-alone, client/server or web-based; extensibility, and storage of ontology data.

Interoperability: a review of capability to interact with other development tools and languages. The four editors reviewed supports merging features, import and export to and from various ontology languages in a range of formats such as XML IDL, KIF, RDF (S), XML(S), OIL, DAML, RDF OCM, OWL, CLOS, Clips, and UML.

Inference Engine: the selected tools are designed with constraint, consistency checking mechanisms, and exception handling. Protégé is the only tool from those analyzed that has a built-in inference engine, KAON2 uses exception handling and others are designed with external inference engines.

Editor usability: addresses ability of this tool to collaborate with other ontology editor's library, versioning and visualization. This study suggests the need for more features and to improve available ones such as edition, help support, and visualization to ensure successful collaboration in ontology construction.

Process capturing & modeling is the ability of ontology tool to capture entire crop processes, specifically for crops like cassava.

In summary, the comparison table presents different properties and functionality used in this analysis. A YES is scored where one or more functions are applicable and an N/A where not applicable. The analysis resulted in the fact that none of these editors are suitable for capturing entire crop process, particularly, knowledge of Cassava Plantation cannot be modeled, which necessitated the need for a flexible editor that can target knowledge engineering.

V. ESSENTIAL FEATURES of REVIEWED ONTOLOGY EDITORS

5.1 Protégé 2000

Essential features include: Import format for XML, XML Schema and RDF(S), Export format for XML, XML Schema, RDF(S), CLIPS, FLogic and Java HTML. Graph view format using Jambalaya plug-in for nested graph view, GraphViz plug-in for browsing classes and global properties.

Consistency checks thru plug-ins using PAL and FaCT, Protégé designed is designed with limited multi-user capabilities, which enables multiple users' interacting with the same database, executing incremental changes without conflict. However, simultaneous changes to the similar data will cause unwarranted problems since there is no support for multiple system users modifying same elements. Protégé provides Web support through OWL plug-in without direct support for Web knowledge base, with the use of servlets; these knowledge bases can be accessed.

In addition to Extensible plug-in architecture, storage capacity, Database and File, one added the advantage of Protégé is that t allows users to browse the knowledge bases without installing the Protégé application.

5.2 Apollo

Essential features include: Import/export format for OCML and Common LISP Object System (CLOS) and does not support graphical view. Inconsistency, Apollo's object model feature allows for robust typing, which enable value check during editing for precise type and existence. Apollo features do not allow undefined instances and classes, neither can you create instances of such classes nor edit their slots, and unclear instances are immediately discarded from the ontology when no reference is made to these instances by any slot. Apollo promises support for weak typing, Metaclasses, support for multi-user, extensible plug-ins, ontology storage and library, all in the future.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

5.3 KAON

Essential features include Import/export format for Resource Description Framework Schema RDF(S) and does not support the graphical view, have internal consistency check and Web support thru KAON Portal. KAON provide a multi-user support, it enables transaction oriented locking, rollback and concurrent access control. Additionally, KAON features allow scalable and efficient reasoning, Meta-modeling comparable to F-Logic via axiom patterns, extends RDFS with symmetric, transitive and inverse relations

5.4 SWOOP

Essential characteristics include Import and Export format for Resource Description Framework Schema RDF(S), OIL, DAML+OIL, SHIQ, dotty and HTML. SWOOP does not support the graphical view. Consistency checks capability thru built-in FaCT. SWOOP has limited web support for RDF URIs, namespaces, and inadequate XML Schemas. In-addition, SWOOP features arbitrary class expressions, which could be used as slot fillers; Concrete type expressions that are not adequately supported; Primitive & defined classes; XML Schema types; Storage and File without extensibility.

5.5 WebOnto

Essential features include Import and Export format for RDF, GXL, RDF(S) and OIL, Web support, Graphical view with little consistency check and multi-user capabilities. Also, WebOnto is designed with Multiple inheritance and exact coverings; a Global write-only locking with change notification; Online service, Ontology Storage and File; Built-in inference engine, Collaborative environment, Meta-classes; Class level support for Prolog-like inference and Information extraction using MnM. WebOnto does not support merging and extensibility.

5.6 OntoLingua

Essential feature includes Import and Export format for IDL, KIF, CLIPS, OKBC and PROLOG syntax. Supports limited consistency check using Chimaera. Provides free web access, storage and files. OntoLingua also provides Multi-user support by write-only locking and user access levels. However, OntoLingua does not support graphical view and no extensibility.

In this section, we have described the essential features of the above ontology tools; Protégé, Apollo, KAON, SWOOP, WebOnto and OntoLingua. Each of these tools is for ontology development. To complete the study, and for the purpose of this research, we choose tools of similar use, for comparison of these tool's features in regard to process capturing.

It should be noted that many other Ontology tools serve a different purpose. For example, PROMPT FCA-merge and Chimaera, are ontology merge and integration tools; COHSE, AeroDAML, OntoAnnotate and MnM are ontology annotation tools. Redland, Sesame, rdfDB, Inkling, cerebra and Jena are ontology storing and querying tools.

We analyze several important aspects of these tools such as the capability of import and export format; the graphical view, multi-user support, extensibility, merging, consistency check, web support, ontology library support and storage, etc. The majority of the Ontology tools reviewed are moving towards Java platforms and extensible architecture, ability to capture entire plantation process, Interoperability and data storage remains the weak point of all these tools.

In conclusion, we have studied some of the advantages and disadvantages of these tools as it relates to entire Cassava process gathering. We conclude that none of these tools have the necessary features to capture entire cassava plantation process.

VI. CONCLUSION & FURTHER RESEARCH

This paper reviewed and analyzed the deficiencies of some of the popular ontology editors and proposed a need for a new ontology editor, capable of capturing entire crop processes. In the final analysis, we can

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

extract the following conclusions. There is no ontology editor designed for agricultural process capturing. Since there is no crop process ontology editor, attempts to modify existing editors is rather complicated in the ontology construction task. In fact, there are many ontology building tools available; most of these editors focus on particular and a few different activities of the ontology lifecycle design; such as editing, documenting, importing /exporting for the various formats, graphical views, ontology libraries, inference engines and browsing functionalities. In conclusion, none of these editors with similar functions can serve the purpose of process capturing. Thus, there is a necessity for a new ontology tool that would capture entire crop process, similar in complexity to cassava plantation.

For continue research, process capturing, merging tools, databases, interoperability with other ontologies/editors, language translations, storage and backup management are an essential improvement in Ontology editor development to avoid additional challenges and improve user experiences.

REFERENCES

[1] J. E. Okeke (1998). Nigeria Cassava Production: National Root Crops Research Institute, Umuahia, Abia State

[2] R. Gruber (1993) Formal Ontology Conceptual Analysis & Knowledge Representation. "Towards principles for the design of ontologies used for knowledge sharing" in Conceptual Analysis and Knowledge Representation.

[3] L. Stojanovic and B. Motik (2002) "Ontology editors evolution within Ontology" in Conference on the Evaluation of Ontology-based Tools

[4] M. Fernandez-Lopez., A. Gomez-Perez and M. D. Rojas-Amaya (2002) Ontologies crossed life cycles, Proc. 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW00), Lecture Notes in Artificial Intelligence, vol. 1937, Springer, Berlin, 2000, pp. 65–79.

[5] D. L. McGuinness and W. Wu (2001). Industrial strength ontology management Semantic Web Working Symposium (pp. 17-37)

[6] N. F. Noy and D. L. McGuinness (2001) "New Ontology Design 101: Guide to Creating Your First Ontology", Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory

[7] M. Koss (2002). *Apollo Editor. Knowledge Media Institute of Open University, United Kingdom*

[8] R. Volz., D. Oberle, S. Staab and B. Motik (2003) KAON SERVER: A semantic Web management system. Alternate Track Proceedings of the Twelfth International World Wide Web Conference, WWW2003 (pp. 20-24).

[9] J. Hendler., A. Kalyanpur., E. Parsia., E. Sirin and Cuenca-Grau (2006) A Web Ontology editing browser *Journal of Web Semantics,* 2006

[10] J. Domingue., E. Motta and O. Garcia Corcho (1999) "OCML Knowledge Modelling in WebOnto: A User Guide", Retrieved from: http://kmi.open.ac.uk/projects/webonto/user_guide.2.4.pdf

[11] J. C. Arpırez., O. Corcho., M. Fernandez-Lopez and A. Gomez-Perez (2001) WebODE: Scalable ontological engineering workbench: First International Conference on Knowledge Capture (KCAP01), ACM Press, Victoria, pp. 6–13

[12] Fikes, R., Farquhar, A., & Rice, J. (1997) Editor for assembling modular ontologies in Ontolingua. AAAI/IAAI (pp.436-441)

https://sites.google.com/site/ijcsis/
ISSN 1947-5500