Abstract: Ontology’s have recently received popularity in the area of knowledge management and knowledge sharing, especially after the evolution of the Semantic Web and its supporting technologies. Ontology defines the terms and concepts (meaning) used to describe and represent an area of knowledge. The aim of this paper is to identify all possible existing ontologies and ontology management tool (Protégé 3.4, Apollo, IsaViz & SWOOP) that are freely available and review them in terms of: a) interoperability, b) openness, c) easiness to update and maintain, d) market status and penetration. The results of the review in ontologies are analyzed for each application area, such as transport, tourism, personal services, health and social services, natural languages and other HCI-related domains. Ontology Building/Management Tools are used by different groups of people for performing diverse tasks. This paper is discussed to ontologies and one of the ontologies tool Protégé

Keywords: Semantic/intelligent Web, Ontology’s, Ontology building tools, Protégé 3.4.

1. INTRODUCTION

Semantic Web [1] is intended to guide the current web to a place where it is more useful for human consumption. It contributes several mechanisms that can be used to classify information and characterize its context for intelligently retrieving information on web. This is mainly done using knowledge representation languages that create explicitly domain conceptualizations, such as ontologies [2],[3]. This conceptualization consists of a set of concepts, their definition and the relationships between them. In recent years, much progress has been made in developing ideas and tools to enable the growth of ontologies. Here, Ontology is widely viewed as the backbone to support various types of information management including information retrieval, storage, and sharing on web. The development of ontologies demands the use of various software tools [4]. A range of open-source and commercial tools are available which assist in the development of various ontologies called Ontology Editors. These tools can be applied to several stages of the ontology life cycle including the creation, implementation, and maintenance of ontologies. This paper emphasis on the role of ontology editor tools for the ontology construction and presents a case study.

![Figure1: Realization of Current Web to Future Web [1]](image-url)

Ontologies are becoming the corner stone of the Semantic Web. Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain. They are shared conceptualizations of a domain and they possibly include the representations of these conceptualizations[5]. They are used to facilitate efficient exchange of information among people, now used for communication among software agents. Ontologies are independent from the applications that use them. This leads to easier software and knowledge maintenance, and contributes to the semantic interoperability between applications. Today a variety of developing environments exist for building ontologies like Protégé 3.4, IsaViz, Apollo, and SWOOP[6],[7],[8],[9]. Protégé 3.4 [6] is a knowledge based ontology editor providing graphical user interface. It is chosen because it provides better flexibility for meta-modeling, enables the construction of domain
ontologies; customize data entry forms to enter data. It is typically targeted at the knowledge engineering and conceptual modeling without knowing or thinking about syntax of output language.

2. METHOD FOR BUILDING ONTOLOGY

This section presents, in direct chronological order, the most well-known approaches for building ontologies [10] from scratch, as well as reusing ontologies that are stored in ontology libraries. First the main set of criteria used to compare different approaches of this type is presented. Then, a brief description of each approach is provided, presenting who has elaborated it and the proposed steps and activities. There is no one correct methodology for developing ontologies (See the appendix ‘A’ for comparison of all methods). Developing ontology is usually an iterative process. We can start with a rough first pass at the ontology and then revise and refine the evolving ontology. Ontology is a model of a real domain in the world and the concepts in the ontology must reflect this reality. After defining an initial version of the ontology, we can evaluate and debug it by using it in applications or problem-solving methods or by discussing it with experts in the field. As a result, we will almost certainly need to revise the initial ontology. This process of iterative design will likely continue through the entire lifecycle of the ontology.

Developing Ontology may include:
• Selection of Domain and Scope
• Consider Reuse
• Find out Important Terms
• Defining Classes and Class Hierarchy
• Defining Properties of Classes and Constraints
• Create Instances of classes

To construct an ontology one must have an ontology specification language, of which there are several to choose. Among many ontology languages, the Web Ontology Language (OWL) is the widely accepted standard for representing and sharing knowledge in the Semantic Web context.[11] OWL is based on Resource Description Framework (RDF) and the DARPA Agent Markup Language (DAML) [12]. OWL would use the RDF meaning of classes and properties (rdfs: Class, rdfs: subclassOf, etc.) and would add some very powerful modeling primitives to extend the expressiveness. OWL also provides an owl: imports construct which syntactically includes the complete referenced ontology into the importing ontology [13],[14]. This construct does not allow partial reuse but can only handle complete ontologies. When starting out on an ontology project, the first and reasonable reaction is to find a suitable ontology software editor. These tools can help acquire, organize, and visualize the domain knowledge before and during the building of a formal ontology. Ontologies on the Web require more expressiveness. Classes are the focus of most ontology. Classes describe concepts in the domain. Slots describe properties of classes and instances. Developing ontology includes [15]:
1. Defining classes in the ontology.
2. Arranging the classes in a taxonomic (subclass–superclass) hierarchy.
3. Defining slots and describing allowed values for these slots.
4. Filling in the values for slots for instances.

3. EVALUATION FRAMEWORK FOR METHODOLOGIES AND METHODS

The set of criteria used to compare different approaches is based on the framework which adapts in [16] which adapts the IEEE 1075-1995 standard for software development process [17]. Such process is broken down in other processes (management processes, development-oriented processes, etc.). Processes are made by activities. For each activity of the framework, we set up whether it is proposed or not, and if it is described in detail. The following kinds of processes are distinguished:

• Project management processes. They create the framework for the project and ensure the right level of management throughout the entire product life cycle. Activities related to project initiation (participants, scheduling, etc.), project monitoring and control, and ontology quality management belong to this group of processes.

• Ontology development-oriented processes. Produce, install, operate and maintain the ontology and it from use. They are divided into three groups:

  - Pre-development processes. They are performed prior to the actual ontology development. They involve activities related to the study of the ontology installation environment, and to feasibility studies.

  - Development processes. These are the required processes for building the ontology. They include: requirements, which are comprised of iterative activities directed towards developing the ontology requirements specification; design process, the goal of which is to develop a coherent and well-organized representation of the ontology that meets the requirements specification of an ontology into an implementation language.

  - Post-development processes. They are related to the installation, operation, support, maintenance and
retirement of an ontology. They are performed after the ontology construction.

- **Integral processes.** These processes are needed to successfully complete ontology project activities. They ensure the completion and quality of project functions. They are performed at the same time as ontology development-oriented processes and include activities that do not output ontology, but are absolutely necessary to obtain a successful system.

At present the construction of ontologies is very much an art rather than a science [18]. The attempt is to formalize the ad-hoc process consists of the some basic steps [19]. A tool that supports methodology and to guide users step by step through the ontology engineering process an effective tool is desired. Along with the development of the methodology we therefore extended the core functionalities of Protégé.

### 4. ONTOLOGY BUILDING TOOLS

In this section, with reference to a Survey1 published on XML site that covers Software Tools that have Ontology editing capabilities and are in use today. These ontology building tools (i.e. Protégé 3.4, IsaViz, SWOOP and Apollo) may be useful for building ontology schemas (terminological component) alone or together with instance data. Concise descriptions of each software tool were compiled and then reviewed by the organization currently providing the software for commercial, open, or restricted distribution. The descriptions are factored into a dozen different categories covering important functions and features of the software.

We have used only “popular and accepted” ontology authoring tool (Protégé 3.4), taking into consideration the advantages of these tool. Tool that provide support for the different phases of the ontology engineering process are referred to as ontology building tools. These tools are used for building a new ontology either from scratch or by reusing existing ontologies, which usually supports editing, browsing, documentation, export and import from different formats, views; libraries and they may have attached inference engines, etc. [20]. The ontology editors are tool that allow users to visually manipulate, inspect, browse and code Ontologies and support in this way the ontology development and maintenance task [21]. In this section, we will provide a broad overview of some of the available ontology editor tools with a brief description of each tool, presenting the group that has developed it, its main features and functionalities, its URL etc.

#### 4. PROTÉGÉ

Protégé [6] is an ontology and knowledge base editor produced by Stanford University. Protégé is a tool that enables the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the properties of these relationships. Protégé is free and can be downloaded from http://protege.stanford.edu [6]. Protégé comes with visualization packages such as OntoViz, EZPal, etc.; all of these help the user visualize ontologies with the help of diagrams. Stanford University is doing a magnificent job of continually improving Protégé. As part of its last update, Protégé now includes an interface for SWRL (Semantic Web Rule Language), which sits on top of OWL to do math, temporal reasoning, and adds Prolog-type reasoning rules. Stanford has a tutorial that covers the basics of using Protégé with the OWL plug-in. The strength of Protégé is that it supports at the same time tool builders, knowledge engineers and domain specialists. This is the main difference with existing tools, which are typically targeted at the knowledge engineer and lack flexibility for meta-modeling. This latter feature makes it easier to adapt Protégé to new requirements and/or changes in the model structure.

When starting out on an ontology project, the first and reasonable reaction is to find a suitable ontology software editor [4]. These tools can help acquire, organize, and visualize the domain knowledge before and during the building of a formal ontology. From the existing tools Protégé 3.4 is chosen because it enables the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables,
variable-value restrictions, and the relationships between classes and the properties of these relationships.

**Figure 3:** Protégé 3.4 screenshot

5. CONCLUSION AND FURTHER RESEARCH

For Ontology development, effective tools are a central requirement. Fortunately, software tools are already available to achieve most of the required activities of ontology development allowing us to focus specifically on the innovative requirements of ontology development within the model. Projects often involve solutions using numerous ontologies (Wine.rdf, food. Owl, Companies.rdf etc.) from external sources. Sometimes there is also the need to use existing and newly developed in-house ontologies (i.e. camera. Owl). For this reason it is important that the editing tools for ontology construction promote interoperability. Ontology Editors prove an asset in the development of ontologies. The need is to identify a suitable editor for a particular domain. A theoretical attempt has been made to analyze and make a comparative analysis of the various ontology editors available and their role in ontology building and maintenance. It can be further extended to choose and make use of an ontology editor for a particular domain ontology creation. To conclude, there are open source ontology tool (Protégé 3.4.Some tools only support common edition and browsing functionalities. Other tools provide ontology documentation, ontology import/export for different formats, graphical view of ontologies, ontology libraries and attached inference engines. It is quite clear Ontology development is an ad-hoc approach. Among several viable alternatives, one need to find which one would work better for the projected task that can easily and effectively be maintained and expressed. Though foundation of ontology is logic but it is a model of reality and the concepts in the ontology must reflect this reality. We have described a tool-assisted method for building the basis for ontologies adopted from domain analysis.

**REFERENCES**


