Developing Ontology for Smart Irrigation of Vineyards

Chapter · January 2019

DOI: 10.1007/978-981-13-5953-8_2

2 authors:

Archana Chougule
Annasaheb Dange College of Engineering and Technology Ashta

Debajyoti Mukhopadhyay
Bennett University

Developing Ontology for Smart Irrigation of Vineyards

Archana Chougule and Debajyoti Mukhopadhyay

Abstract As the availability of groundwater is getting reduced these years, proper scheduling of irrigation is very important for survival and improvement of vineyards in the southern part of India. Well calculated irrigation scheduling also improves quality of grapes. The knowledge about irrigation to vineyards is available in various documents, and it is scattered. This knowledge can be made available to grape growers through computer-based applications. As Semantic Web is playing an important role, information sharing among varying automated systems, extraction of information from such documents and representing it as ontology is a good idea. This paper presents techniques for automated extraction of knowledge from text resources using natural language processing technique for building vineyard ontology. Smart irrigation system can be developed using IoT sensors and other ICT devices. The paper explains ontology built for resources used under smart irrigation system. It suggests how smart irrigation systems can be built by grape growers by utilizing vineyard ontology and smart irrigation ontology.

Keywords Ontology building · Irrigation scheduling · Grapes · Knowledge base · Natural language processing

1 Introduction

Water is the most important resource in agriculture. Irrigation scheduling is basically deciding on the frequency and duration of water supply to any crop. Proper scheduling of irrigation can minimize wasting water and can also help to improve the quality of grapes. There are different methods of irrigation. Drip irrigation is used by almost all vie yards in India. Manerajuri is a village in India, which is famous for grapes, but

A. Chougule (✉)
Walchand College of Engineering, Sangli, India
e-mail: chouguleab@gmail.com

D. Mukhopadhyay
NHITM, Mumbai University, Mumbai, India
e-mail: debajyoti.mukhopadhyay@gmail.com

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_2

more than 10,000 bore wells have run dry in year 2016. This fact motivated authors to develop the knowledge base and decision support system which can be used for micromanagement of available water.

Knowledge base generation for scheduling water supply to grapes in hot tropical region in India will help automation of water scheduling. Good irrigation practices in ontology will be good education material for farmers. Structured representation of knowledge is more useful than unstructured one. Ontologies can be used for structured representation of concepts in any domain. Along with concepts, relationships between concepts can also be mentioned using ontology. Maintenance and sharing of information can be facilitated using ontologies. Manual construction of domain ontology is time-consuming and error-prone. Semi-automated approach will help in enriching ontology building, reducing required time and enhancing the quality of built ontology. For IoT-based automation of irrigation systems, formal representation and management of IoT techniques and devices used important.

Automated irrigation system needs information like ambient temperature and humidity, atmospheric pressure, soil temperature, soil moisture and leaf wetness. System can be built with such information using sensors, actuators, monitors, collectors and transmitters. Data generated and required from such varying sources can be easily integrated using ontology [1].

This paper describes building ontology for developing IoT-based automated irrigation systems. It details mechanism used for constructing ontology. Two types of ontologies, namely, vineyard irrigation ontology and smart irrigation ontology, are described. These ontologies can be used for generating knowledge base and establishment of automated irrigation system based on available resources. It can also be used to educate grape growers about principles of vineyard irrigation.

2 Research Method

Building ontology consists of five basic steps as defining domain, scope and objective of ontology, listing important keywords from the domain, finding hierarchies between keywords, defining relations between them and the last step is to add those keywords as concepts in ontology and define relationships between concepts. Various formats for building ontology are available as RDF, RDF-S, OWL and OWL-DL. Here ontology is developed in Web Ontology Language (OWL) format. Classes, individuals, data properties, object properties and axioms are the main parts of OWL ontology. Classes represent core concepts in the domain. Individuals are instances of any specific class. Data properties are attributes of a class with specific values and object properties hold values as objects of some other class. Two ontologies are maintained separately as vineyard ontology and smart irrigation ontology.

2.1 Building Vineyard Irrigation Ontology

Water management is also important to maintain yield and quality of grapes. This section details how water management knowledge is represented in ontology form. Knowledge about irrigation scheduling of grapes is available in various documents published by researchers from organizations like National Research Center for Grapes. To make use of this knowledge in decision support system, it must be transformed to accessible form. As mentioned earlier, knowledge stored in ontology can be accessed and shared by automated systems. This section discusses converting the irrigation scheduling details available in text documents to ontology. As a reference ontology, AGROVOC [2] is used. AGROVOC is an agricultural vocabulary available as ontology. It is available in resource description framework (RDF) format.

The documents for irrigation scheduling are taken as input and the most relevant documents are only considered. Relevancy of documents is ranked using TF-IDF [3] algorithm. TF-IDF stands for term frequency, inverse document frequency. List of keyword related to agriculture taken from agriculture experts is considered as input for this purpose. Term frequency is considered as count of word occurring in document and inverse document frequency is the count of documents containing the word versus the total number of documents. Natural language processing techniques are used then to extract important keywords from the text. The text is converted to a number of tokens using StringTokenizer. The text is broken into words, phrases and symbols under tokenization. Tokenization helps in finding meaningful keywords from the text.

After tokenization, stop words are removed from the list of tokens. Stop words are words which are used for joining words together in a sentence. They occur very frequently in documents, but are of no importance for ontology building. Stop words like ‘and’, ‘or’, ‘this’ and ‘the’ are not used for the classification of documents, so they are removed. The process of conflating the variant forms of a word into a common representation is called stemming. There can be the same words with different forms. Such words are converted to common form using Porter’s stemmer [4]. For example, the words: ‘irrigation’, ‘irrigated’ and ‘irrigating’ are reduced to a common representation ‘irrigate’. OpenNLP [5] POS tagger is used for extracting important keywords from tokens. POS tagger assigns labels to keywords as NN for noun and NNP as proper nouns. Assigned labels are then considered for probable classes, data properties, individuals or object properties.

Along with concepts relationships between concepts are also found by processing sentences. Formal concept analysis [6] is used for deciding hierarchy among concepts. Formal concept analysis is a technique used for defining conceptual structures among data sets [7]. Extracted individuals, data properties and object properties are used for this purpose. The top-down approach is used here. Extracted concepts are stored in the concept table. Extracted relations are used for defining hierarchy among classes and among properties. For example, Table 1 has five concepts and four attributes, showing which concepts carry which corresponding attributes, and Fig. 1 shows corresponding concept lattice.

Table 1 Example of formal concept analysis

	Amount	Unit	Timestamp	Level
Sensor		X	X	
Irrigation	X	X	X	
Groundwater	X	X	X	X
Surface water			X	X
Soil moisture		X	X	X

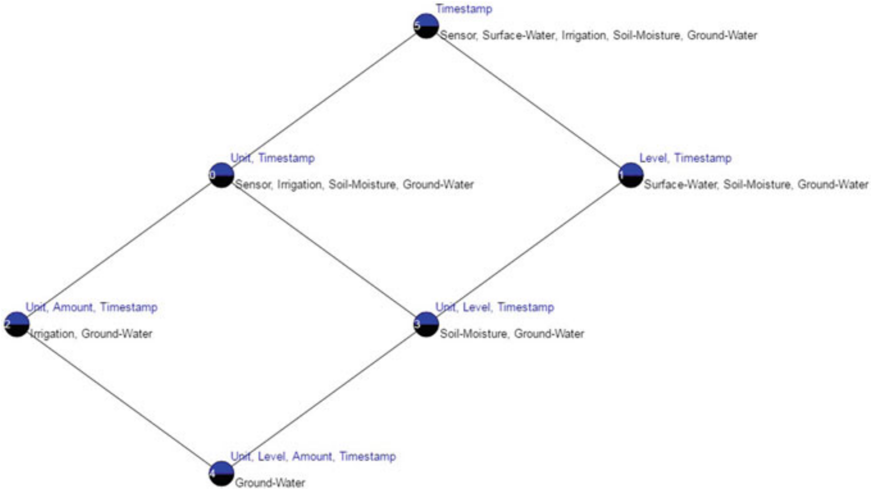

Fig. 1 Concept lattice corresponding to context in Table 1

The ontology is built using ProtegeOWL APIs [8] available in java language. The fundamental concepts related to irrigation are added as classes in ontology. The ontology is built using the top-down approach. More general term is added at higher level in the hierarchy, followed by more specific concepts at lower levels.

Vineyard ontology contains 16 classes, 56 individuals, 12 object properties and 19 data properties. The built ontology is further edited using Protégé 5.1 ontology editor. Protégé editor has facilities to add, edit and delete concepts, change hierarchy, view ontology and use reasoners on ontology. The irrigation knowledge is extracted from research documents published by national research for grapes, Pune, India [9]. Irrigation schedule can be decided based on knowledge stored in vineyard irrigation ontology in terms of individuals, object properties and data property values. Requirements for irrigation for grapes are mentioned as individuals of Irrigation-Schedule class. Under each object, water requirement is specified as data property value. Growth stage-wise water requirements of grapes are considered and mentioned for each irrigation schedule.

Figure 2 shows classes under vineyard ontology. Water-Shortage-Symptoms class stores all symptoms shown on grape leaves, stems, berries and in soil due to the

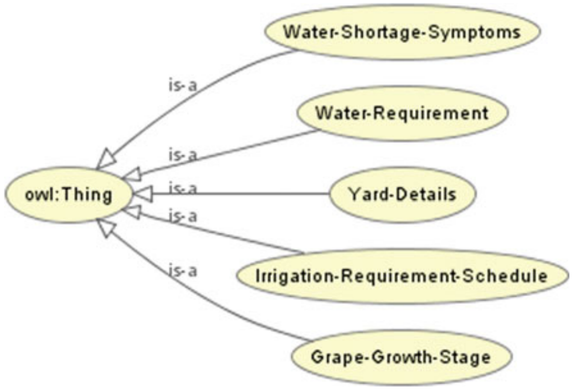

Fig. 2 Classes in vineyard ontology

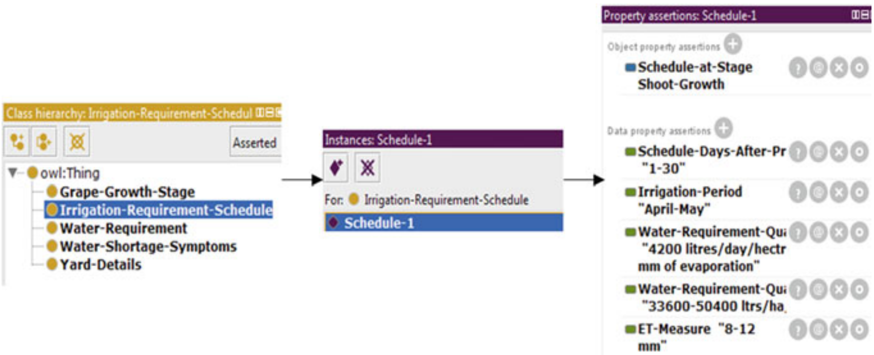

Fig. 3 Example of irrigation schedule defined in vineyard ontology as data and object property assertions

shortage of water. Specific information about yard like root depth, soil type and available water resources are covered by Yard-Details class. All stages of vine growth are stored as individuals of Grape-Growth-Stage class and are referenced by irrigation requirement schedule.

Irrigation-Requirement-Schedule is the most important class for modelling and predicting irrigation requirements. It contains water requirement schedules as individuals detailing days after pruning of schedule, water requirement quantity in litres per day per hectare, growth stage of vines, irrigation period in terms of month and evapotranspiration measure. Figure 3 shows the example of irrigation schedule defined in vineyard ontology.

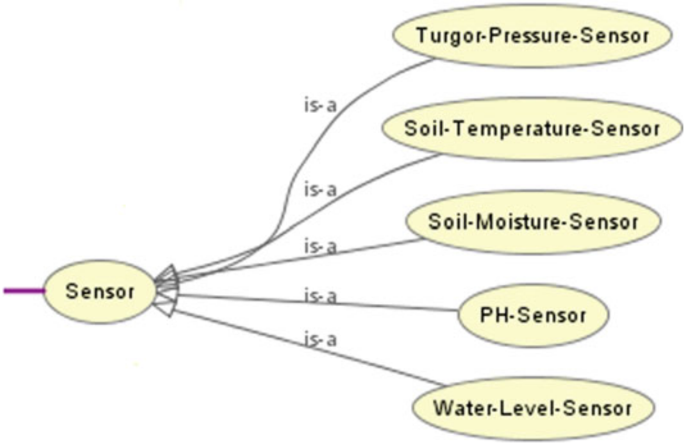

Fig. 4 Sensor class in smart irrigation ontology

2.2 Smart Irrigation Ontology

In order to provide a knowledge base of tools and techniques that can be used for automation of irrigation scheduling, Smart Irrigation ontology is built. It contains all concepts including IoT, for creating and managing smart irrigation system for vineyard. The time and amount of water to apply to vineyard are generally based on four methods as monitoring of soil moisture levels, measuring water status in plants, based on evapotranspiration measurement and based on the estimated vineyard water use rate and soil water storage. As irrigation decision depends on various factors in vineyard, sensors are used for such measurements. Figure 4 shows types of sensors that can be used for smart irrigation. Unit-of-measure, type-of-sensing, locating-of-sensing and sensor-description are data properties of the Sensor class. The ontology provides all support for storing information about IoT-based tools. It also provides data and object properties for storing measurements read by such tools. It comes under devices and communication techniques classes and their sub-classes. Knowledge about irrigation techniques that can be used under vineyards and possible water resources comes under Irrigation-Techniques and Water-Resources classes. Observations class contains sub-classes, data and object properties for storing measurements taken by sensors. System-Design and User-Interface are the classes suggesting options for building irrigation automation system using available resources. As irrigation management depends on climate and weather details in specific region two classes, namely, Climate and Weather-Details, are added in the ontology. Annual precipitation and annual rainfall are data properties of climate class. Rainfall, humidity and temperature are parts of Weather-Details class. Figure 5 shows all classes under smart irrigation ontology.

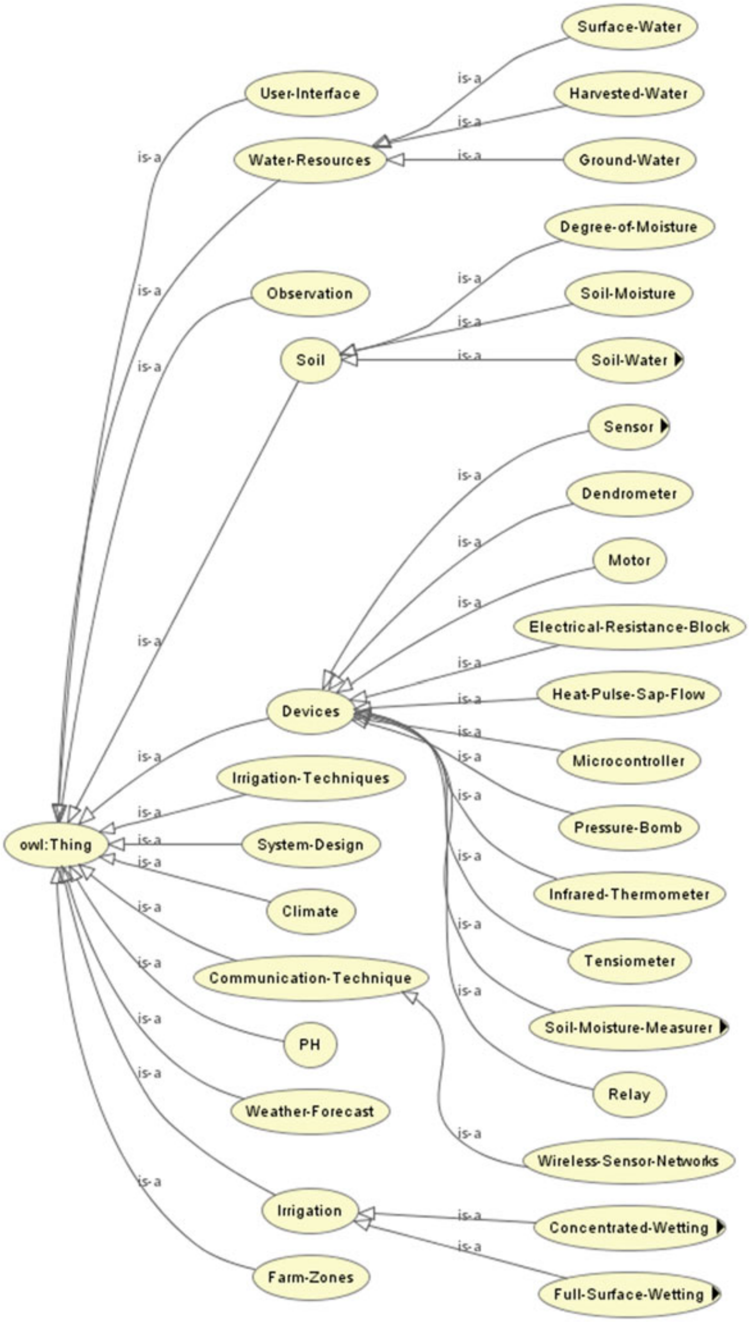

Fig. 5 Classes in smart irrigation ontology

3 Conclusion

Ontology plays an important role in IoT-based automation of agricultural systems. The paper demonstrated how ontology can be built by using natural language processing techniques and formal concept analysis. Paper demonstrated how irrigation details about vineyards can be very well represented using ontology. The information about sensors can also be stored in ontology and used for automated systems building, which is shown in smart irrigation ontology. Based on given water requirements, the automated irrigation system can be built using knowledge from vineyard ontology and smart irrigation ontology as proposed in the paper.

References

1. Cornejo, C., Beck, H. W., Haman, D. Z., & Zazueta, F. S. (2005). Development and application of an irrigation ontology. *2005 EFITA/WCCA Joint Congress on IT and Agriculture*, Vila Real, Portugal, 25–28 July 2005.
2. AGROVOC Thesaurus. http://aims.fao.org/agrovoc#.VF29AvmUc2U.
3. TFIDF Algorithm. http://en.wikipedia.org/wiki/Tf-idf.
4. Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.
5. OpenNLP. https://opennlp.apache.org/.
6. Obitko, M., Snasel, V., Smid, J. (2004). Ontology design with formal concept analysis. In V. Snasel & R. Belohlavek (Eds.), C*LA 2004*, pp. 111–119, ISBN 80-248-0597-9. VSB—Technical University of Ostrava, Deportment of Computer Science.
7. Ganter, B., & Wille, R. (1999). *Formal concept analysis, mathematical foundation*. Berlin: Springer Verlag.
8. Protege-OWL API Programmer's Guide. https://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide [Accessed on 17 May 2017].
9. National Research Centre for Grapes. http://nrcgrapes.nic.in.