

Development of Semantic Web applications: state of art and critical review

Salah-Ddine Krit, Mohamed El Asikri, Brahim El Habil, Hassan Chaib, Mohamed Benaddy, Mourad Elouali

Department Mathematics and Informatics and Management,
Laboratory of Engineering Sciences and Energy.
Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University
Agadir BP/638 Morocco
meda.asikri@gmail.com
salahddine.krit@gmail.com
hchaib@gmail.com

ABSTRACT

Ontologies 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. An ontology defines the terms and concepts (meaning) used to describe and represent an area of knowledge. The aim of this paper is to identify the basics notions of ontologies and ontology management tools specially protégé that are freely available and the process development of semantic web notions.

The Semantic Web vision have lead to a number of standards such as OWL and Web Semantic Mining. While these standards provide a technical infrastructure, developers have little guidance on how to build real-world Semantic Web applications. We will start by some definitions of Semantic Web and the common points with Web Mining. We will illustrate how these notions can be put into practice using the modern Semantic Web development tool Protege, and indicate future possibilities;

Keywords— Semantic Web, Web Mining, ontology, Knowledge discovery, ontology learning.

CCS Concepts

Information systems→Data Mining Web Semantic Mining

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICEMIS'18, June 19–21, 2018, Istanbul, Turkey.

Copyright 2018 ACM ISBN 978-1-4503-6392-1/18/04...\$15.00.

DOI: <https://doi.org/10.1145/3234698.3234753>

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.

Semantic Web is a recent initiative inspired by Tim Berners-Lee [1] and it is an extension of the current Web that provides an easier way to find, share, reuse and combine information. It is based on machine-readable information and builds on XML technology's capability to define customized tagging schemes and RDF's [2] (Resource Description Framework) flexible approach to representing data. The Semantic Web provides common formats for the interchange of data. It also provides a common language for recording how data relates to real world objects, allowing a person or a machine to start off in one database, and then move through an unending set of databases of knowledge in order to find various concepts and words to build ontologies and representation in a way that can be understood and consumed by every users [3,16,17,18,19,20]

Protégé 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.

2.1. Ontology

We start by introducing the following example fig 1 for an ontology and its representation.

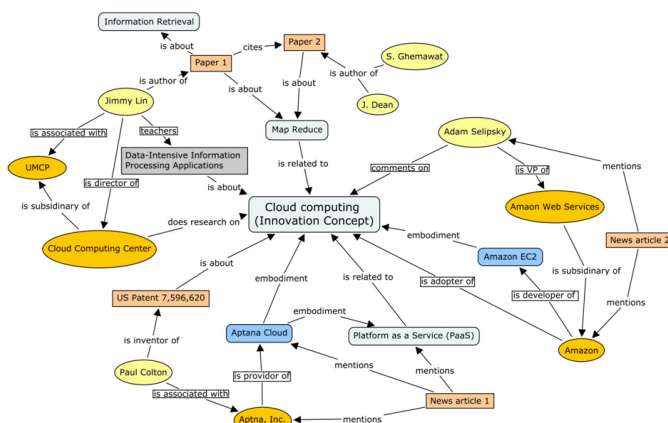


Figure 1 : Ontology Instance

2.2. Semantic web mining

As the Semantic Web enhances the first generation of the WWW with formal semantics, it offers a good basis to enrich Web Mining: The types of (hyper)links are now described explicitly, allowing the knowledge engineer to gain deeper insights in Web structure mining; and the contents of the pages come along with a formal semantics, to apply mining techniques which require more structured input.

2.2.1.Semantic Web Content and Structure Mining

There are two big scientific challenges in this attempt. The first is the size of the data to be processed (i.e. the scalability of the algorithms), and the second is the fact that the data are distributed over the Semantic Web, as there is no central database server. Scalability has always been a major concern for ILP algorithms. With the expected growth of the Semantic Web, this problem increases as well. Therefore, the performance of the mining algorithms has to be improved, e.g. by sampling.

2.2.2. Semantic Web Usage Mining

Usage mining can also be enhanced further if the semantics are contained explicitly in the pages by referring to concepts of an ontology. Semantic Web usage mining can for instance be performed on log files which register the user behavior in terms of an ontology. A system for creating such semantic log files from a knowledge portal has been developed at the AIFB. These log files can then be mined, for instance to cluster users with similar interests in order to provide personalized views on the ontology.

3. THE KNOWLEDGE DISCOVERY PROCESS

3. Transformation The third step produces a projection of the data to a form that data mining algorithms can work on—in most cases, this

means turning the data into a propositional form, where each instance is represented by a feature vector. To improve the performance of subsequent data mining algorithms, dimensionality reduction methods can also be applied in this step to reduce the effective number of variables under consideration.

4. Data mining Once the data is present in a useful format, the initial goal of the process is matched to a particular method, such as classification, regression, or clustering. This step includes deciding which models and parameters might be appropriate (for example, models for categorical data are different than models for numerical data), and matching a particular data mining method with the overall criteria of the KDD process (for example, the end user might be more interested in an interpretable, but less accurate model than a very accurate, but hard to interpret model). Once the data mining method and algorithm are selected, the data mining takes place: searching for patterns of interest in a particular representational form or a set of such representations, such as rule sets or trees.

5. Evaluation and interpretation In the last step, the patterns and models derived by the data mining algorithm(s) are examined with respect to their validity. Furthermore, the user assesses the usefulness of the found knowledge for the given application. This step can also involve visualization of the extracted patterns and models, or visualization of the data using the extracted models.

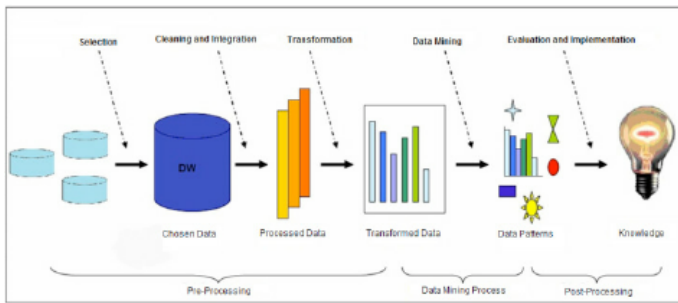


Figure 1 : Process of knowledge discovery

As a means to express knowledge about a domain in the Semantic Web, *ontologies* have been introduced in the early 1990s as “explicit formal specifications of the concepts and relations among them that can exist in a given domain” [6]. For the area of knowledge discovery and data mining, Nigro et al. [7] divide ontologies used in this area into three categories:

- **Domain ontologies:** Express background knowledge about the application domain, i.e., the domain of the data at hand on which KDD and data mining are performed.
 - **Ontologies for data mining process:** Define knowledge about the data mining process, its steps and algorithms and their possible parameters.
 - **Metadata ontologies:** Describe meta knowledge about the data at hand, such as provenance information, e.g., the processes used to construct certain datasets.
- It has been already shown that ontologies for the data mining process and metadata ontologies can be used in each step of the KDD process.

4. EXAMPLE OF USE OF ONTOLOGY-BASED WEB MINING

Ontology-based Web mining, like traditional Web mining, is useful to many different digital library applications. We can group these applications under the following classes:

- **Improved search to Web data :** With additional ontological semantics, Web data can be indexed by their concepts and relationships to support expressive search queries. A more expressive query model can support very precise information search and reduce the amount of irrelevant Web information in the results [13].
- **Better browsing capabilities :** Similar to searching, Web pages can be browsed based on their ontology concepts and relationships instead of following Weblinks only. If Web pages are the concept instances, relationship instances can be created as some virtual links between Web pages. Other than selecting Web pages belonging to concepts of interest, one can thus navigate the virtual links between Web pages enriching the browsing experience in digital library applications[13]. On the other hand, if some text elements are identified as concept instances and their relationships are extracted, they can also be marked up in Web pages to direct user attentions to the more important text passages.
- **Personalization of Web data access :** Personalization aims to find a subset of Web data that matches the interest profile of a user or a group of users. This can be achieved by recommending Web pages or Websites to the user(s), or by filtering Web pages that are of interest to the user(s). For example, this can be done by analysing the historical data recording user accesses to Web data, and mining the topics relevant to a user by clustering previously accessed Web pages based on content similarities. When a new Web page is found to be similar to one of the clusters, it can be routed to the user. As Web pages are annotated with ontology entity labels, the grouping of Web pages accessed by a user can be more effectively done leading to more effective content recommendation.

5. ONTOLOGY LEARNING PROCESS

Ontology learning is used to (semi-)automatically extract whole ontologies from natural language text or Web Pages. The process is usually split into the following eight tasks, which are not all necessarily applied in every ontology learning system[8][9].

- Domain terminology extraction
- Concept discovery
- Concept hierarchy derivation
- Learning of non-taxonomic relations
- Rule discovery
- Ontology population
- Concept hierarchy extension

5.1. Domain terminology extraction

During the domain terminology extraction step, domain-specific terms are extracted, which are used in the following step (concept discovery)

to derive concepts. Relevant terms can be determined e. g. by calculation of the TF/IDF values or by application of the C-value / NC-value method. The resulting list of terms has to be filtered by a domain expert. In the subsequent step, similarly to coreference resolution in IE, the OL system determines synonyms, because they share the same meaning and therefore correspond to the same concept. The most common methods therefore are clustering and the application of statistical similarity measures.

5.2. Concept discovery

In the concept discovery step, terms are grouped to meaning bearing units, which correspond to an abstraction of the world and therefore to concepts. The grouped terms are these domain-specific terms and their synonyms, which were identified in the domain terminology extraction step.

5.3. Concept hierarchy derivation

In the concept hierarchy derivation step, the OL system tries to arrange the extracted concepts in a taxonomic structure. This is mostly achieved by unsupervised hierarchical clustering methods. Because the result of such methods is often noisy, a supervision, e. g. by evaluation by the user, is integrated. A further method for the derivation of a concept hierarchy exists in the usage of several patterns, which should indicate a sub- or supersumption relationship. Patterns like “X, that is a Y” or “X is a Y” indicate, that X is a subclass of Y. Such pattern can be analyzed efficiently, but they occur too infrequent, to extract enough sub- or supersumption relationships. Instead bootstrapping methods are developed, which learn these patterns automatically and therefore ensure a higher coverage.

5.4. Learning of non-taxonomic relations

At the learning of non-taxonomic relations step, relationships are extracted, which do not express any sub- or supersumption. Such relationships are e.g. works-for or located-in. There are two common approaches to solve this subtask. The first one is based upon the extraction of anonymous associations, which are named appropriately in a second step. The second approach extracts verbs, which indicate a relationship between the entities, represented by the surrounding words. But the result of both approaches has to be evaluated by an ontologist.

5.5. Rule discovery

During rule discovery axioms[10] (formal description of concepts) are generated for the extracted concepts. This can be achieved for example by analyzing the syntactic structure of a natural language definition and the application of transformation rules on the resulting dependency tree. The result of this process is a list of axioms, which is afterwards comprehended to a concept description. This one has to be evaluated by an ontologist.

5.6. Ontology population

At the ontology population step, the ontology is augmented with instances of concepts and properties. For the augmentation with instances of concepts methods, which are based on the matching of lexico-syntactic patterns, are used. Instances of properties are added by application of bootstrapping methods, which collect relation tuples.

5.7. Concept hierarchy extension

In the concept hierarchy extension step, the OL system tries to extend the taxonomic structure of an existing ontology with further concepts. This can be realized supervised by a trained classifier or unsupervised by the application of similarity measures.

5.8. Tools

Dog4Dag - an ontology generation plugin for Protégé 4.1 and OBOEdit, ontoStudio fig 3 . DOG4DAG is an ontology generation plugin for both Protégé 4.1 and OBO-Edit 2.1. It allows for term generation, sibling generation, definition generation, and relationship induction. Integrated into Protégé 4.1 and OBO-Edit 2.1, DOG4DAG allows ontology extension for all common ontology formats (e.g., OWL and OBO). Limited largely to EBI and Bio Portal lookup service extensions.[11]

OWL ontology languages [12] allow users to write explicit, formal conceptualization of domain models. They full fill the main requirements of ontology languages such as

- A well-defined syntax
- A formal semantics
- Convenience of expression
- Efficient reasoning support

The subclass relationship between OWL and RDF property OWL builds on RDF and RDF schema and uses RDF,,s XML based syntax. OWL can be defined with XML based syntax, Abstract syntax which makes use of RDF and Graphical Syntax. Proposed architecture applied to generate ontology based on web log which described in web ontology language (OWL).Web authors will be helped by analyzing user “browser” history of web sites which are frequently visited by people and the items which are popular in market and the same is easily identified by this application. Web structure mining has used in developing ontologies and help to retrieving process.

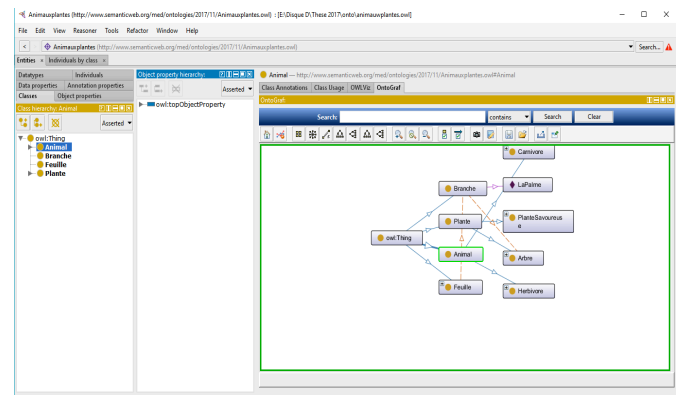


Figure 2 : Building ontolgy example with Protégé

6. ONTOLOGY-DRIVEN SOFTWARE DEVELOPMENT : PROTÉGÉ

The Semantic Web applications consist of two separate but linked layers: The Semantic Web Layer makes ontologies and interfaces available to the public, whereas the Internal Layer consists of the

control and reasoning mechanisms. While the latter components can reside inside a “black box”, the artifacts in the Semantic Web Layer are shared with other applications, and must therefore meet higher quality standards than the internal components. Also, the models in the Semantic Web Layer are used to control the internal behavior, in particular the outcome of reasoning algorithms. As a result, the code inside the internal components may be of relatively small size, and has to meet lower quality standards than the externally visible modules. Much of the internal components can be generated from the higher-level models and consists of generic libraries that are simply used out of the box. All this suggests that agile development approaches such as eXtreme Programming [14] are perfectly sufficient for the development of the application’s internal parts.

Much more effort has to be spent with the system’s ontologies. These ontologies must be consistent, generally useful and formally correct. Also, they must be extensible and may become very large. In order to ensure a high quality of these ontologies, developers can exploit the formal foundation of OWL on Description Logics at design time to find inconsistencies, to maintain complex subclass relationships along multiple axes, and to reveal hidden relationships.

Modern ontology development tools such as Protégé with the OWL Plugin [15] allow users to exploit these services conveniently, and provide intelligent guidance to find mistakes similar to a debugger in a programming environment. As illustrated in Figure 3, the user only needs to press a “classify” button to get the correct classification of classes and to detect inconsistencies. Furthermore, Protégé serves as a rapid prototyping environment in which ontology designers can instantly create individuals of their ontology and experiment with semantic restrictions. The system is able to generate user interfaces that can be further customized for knowledge acquisition in a particular domain.

In addition to supporting ontology designers, Protégé also has an open architecture that allows programmers to insert arbitrary components into the tool. This can be exploited during the development of Semantic Web applications: developers can package the implementation of the application as a Protégé plugin and immediately test how the system behaves in response to ontology changes. The Protégé OWL Plugin even provides an open testing framework in which code similar to JUnit test cases can be executed at any time. Another feature of Protégé is code generation: the system takes an OWL ontology and creates corresponding Java classes from it.

To summarize, the development of Semantic Web applications with a tool such as Protégé is driven by ontologies. Ontologies can be developed by domain experts, and these experts have direct control over the behavior of the executing system and some aspects of the implementation. Similar to agile methodologies like eXtreme Programming, feedback is available frequently since the domain models rapidly lead to executable systems. Best practices from agile methodologies can be employed to build high-quality domain models. In particular, ontologies can be developed in pairs, so that they are instantly peer-reviewed. To ensure that domain models lead to efficient implementations, domain experts may pair with programmers. Another important aspect of agile approaches is systematic testing. The formal backing of languages like OWL can be employed for testing both at design-time and at run-time. For example, the logical definitions from an OWL class can be exploited to verify invariants on objects similar to Design-by-Contract at run-time.

6. CONCLUSION AND OUTLOOK

For Ontology development, effective tools are a central requirement. Software tools are already available to achieve most of the required activities of ontology development allowing us to focus specifically on the innovate requirements of ontology development within the model. Projects often involve solutions using numerous from external sources. Sometimes there is also the need to use existing and newly developed in-house ontologies(i.e. animal.owl). For this reason it is important that the editing tools for ontology construction promote interoperability, in conclusion there are many tools that contribute to build ontology and to give the developers a great help with open source tools and very huge plugins of web Semantic Mining. We expect that, in the future, Web Mining methods will increasingly treat content, structure, and usage in an integrated fashion in iterated cycles of *extracting* and *utilizing* semantics, to be able to understand the Web. Among those iterated cycles, we expect to see a productive complementarity between those relying on semantics in the sense of the Semantic Web, and those that rely on a looser notion of semantics.

7. REFERENCES

- [1] T.Berners-Lee, N. Shadbolt, and W.Hall, “The Semantic Web Revisited” IEEE Intelligent Systems, pp. 96-101, 2006
- [2] <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [3] M.-S. Chen, J. Han, and P. S. Yu, Data mining: an overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6):866-883.
- [4] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R.Uthrusamy, eds., Advances in knowledge discovery and data mining, Menlo Park, California : AAAI Press/ The MIT Press, 1996.
- [5] Web-Ontology (WebOnt) Working Group, 2001, <http://www.w3.org/2001/sw/WebOnt/>.
- [6] T.R. Gruber Toward principles for the design of ontologies used for knowledge sharing Int. J. Hum.-Comput. Stud., 43 (5) (1995), pp. 907–928
- [7] H.O. Nigro, S.G. Cisaró, D.H. Xodo Data Mining With Ontologies: Implementations, Findings and Frameworks, Information Science Reference, Imprint of: IGI Publishing, Hershey, PA (2007)
- [8] Cimiano, Philipp; Völker, Johanna; Studer, Rudi (2006). "Ontologies on Demand? - A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text", Information, Wissenschaft und Praxis, 57, p. 315 - 320
- [9] Wong, W., Liu, W. & Bennamoun, M. (2012), "Ontology Learning from Text: A Look back and into the Future". ACM Computing Surveys, Volume 44, Issue 4, Pages 20:1-20:36.
- [10] Völker, Johanna; Hitzler, Pascal; Cimiano, Philipp (2007). "Acquisition of OWL DL Axioms from Lexical Resources", Proceedings of the 4th European conference on The Semantic Web, p. 670 - 685
- [11] Thomas Wächter, Götz Fabian, Michael Schroeder: DOG4DAG: semi-automated ontology generation in OBO-Edit and Protégé. SWAT4LS London, 2011. doi:10.1145/2166896.2166926.
- [12] <https://www.w3.org/OWL/>

[13] Naing, M.-M, Lim, E.-P., and Chiang, R. H.-L., “Core: A Search and Browsing Tool for Semantic Instances of Web Sites,” Asia Pacific Web Conference (APWeb’05) , 2005

[14] Kent Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading, MA, 1999.

[15] Holger Knublauch, Ray W. Ferguson, Natalya F. Noy, and Mark A. Musen. The Protégé OWL Plugin: An open development environment for semantic web applications. In 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, 2004

[16] Shadi A. Aljawarneh, Muneer Bani Yassein, and We'am Adel Talafha. 2018. A multithreaded programming approach for multimedia big data: encryption system. Multimedia Tools Appl. 77, 9 (May 2018), 10997-11016. DOI: <https://doi.org/10.1007/s11042-017-4873-9>

[17] Vangipuram Radhakrishna, Shadi A. Aljawarneh, Puligadda Veereswara Kumar, and Kim-Kwang Raymond Choo. 2018. A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns. Soft Comput. 22, 6 (March 2018), 1903-1919. DOI: <https://doi.org/10.1007/s00500-016-2445-y>

[18] Shadi A. Aljawarneh, Muneer Bani Yassein, and We'am Adel Talafha. 2017. A resource-efficient encryption algorithm for multimedia big

data. Multimedia Tools Appl. 76, 21 (November 2017), 22703-22724. DOI: <https://doi.org/10.1007/s11042-016-4333-y>

[19] Shadi A. Aljawarneh, Ali Alawneh, and Reem Jaradat. 2017. Cloud security engineering. Future Gener. Comput. Syst. 74, C (September 2017), 385-392. DOI: <https://doi.org/10.1016/j.future.2016.10.005>

[20] Shadi A. Aljawarneh, Radhakrishna Vangipuram, Veereswara Kumar Puligadda, and Janaki Vinjamuri. 2017. G-SPAMINE. Future Gener. Comput. Syst. 74, C (September 2017), 430-443. DOI: <https://doi.org/10.1016/j.future.2017.01.013>

[21] Muneer Bani Yassein, Shadi A. Aljawarneh, and Esraa Masadeh. 2017. A new elastic trickle timer algorithm for Internet of Things. J. Netw. Comput. Appl. 89, C (July 2017), 38-47. DOI: <https://doi.org/10.1016/j.jnca.2017.01.024>

[22] Shadi A. Aljawarneh, Mohammed R. Elkobaisi, and Abdelsalam M. Maatuk. 2017. A new agent approach for recognizing research trends in wearable systems. Comput. Electr. Eng. 61, C (July 2017), 275-286. DOI: <https://doi.org/10.1016/j.compeleceng.2016.12.003>

[23] Muneer O. Bani Yassein and Shadi A. Aljawarneh. 2016. A Conceptual Security Framework for Cloud Computing Issues. Int. J. Intell. Inf. Technol. 12, 2 (April 2016), 12-24. DOI=<http://dx.doi.org/10.4018/IJIT.2016040102>

Authors' background

Your Name	Title*	Research Field	Personal website
Mohamed El Asikri	Phd Student	phD student in Polydisciplinary Faculty of Ouarzazate, Department Mathematics and Informatics and Management, Laboratory of Engineering Sciences and Energy, Ibn Zohr University- Agadir. His research interests Knoledge discovery and Web Semantic Mining .	
Salah-ddine Krit	full professor	Salah-ddine Krit received the B.S. and Ph.D degrees in Microelectronics Engineering from Sidi Mohammed Ben Abdellah university, Fez, Morocco. Institute in 2004 and 2009, respectively. During 2002-2008, he is also an engineer Team leader in audio and power management Integrated Circuits (ICs) Research. Design, simulation and layout of analog and digital blocks dedicated for mobile phone and satellite communication systems using CMOS technology. He is currently a professor of informatics- Physics with Polydisciplinary Faculty of Ouarzazate, Ibn Zohr university, Agadir, Morocco. His research interests include wireless sensor Networks (Software and Hardware), computer engineering and wireless communications.	https://kritsalahddine.blogspot.com
Hassan Chaib	full professor	received his Bachelor's degree (License es Sciences) in Physics in 1994 and his Master's degree (DESA) in physics from the faculty of sciences, university IbnouZohr, Agadir , Morocco, in 1995, and He received his Ph.Din 2000, He joined Polydisciplinary Faculty of	