# Application of Ontology Modularization for Building a Criminal Domain Ontology

Mirna El Ghosh[1], Hala Naja[2], Habib Abdulrab[1], Mohamad Khalil[3]

[1] LITIS, INSA Rouen, France
[2] Faculty of Sciences, Lebanese University, Lebanon
[3] CRSI Research Center, Faculty of Engineering, Lebanese University, Lebanon

**Abstract.** The Ontology modularization is an essential field in the ontology engineering domain helping to reduce the complexity and the difficulties of building, reusing, managing and reasoning on domain ontologies either by applying partitioning or composition approaches. This paper carries out a survey on ontology modularization and presents a modular approach to build criminal modular domain ontology (CriMOnto) for modelling the legal norms of the Lebanese criminal system. CriMOnto, which will be used later for a legal reasoning system, is composed of four independent modules. The modules will be combined together to compose the whole ontology.

**Keywords:** Ontology Modularization, Modular Ontology, Ontology Composition, Ontology Integration, Ontology Reuse, Criminal Domain, CriMOnto.

## 1 Introduction

It is commonly known that ontologies aim to capture consensual knowledge of a given domain in a generic and formal way, to be reused and shared across applications and by groups of people [1]. In this context, the number of available ontologies has increased considerably in various domains such as bioinformatics, genetics, medicine and law, among many others, but they are also becoming larger and more complex to manage and reuse [2]. In this context, interest in modularization techniques, as an ontology engineering principle, has increased to resolve the problems of reusability, scalability and maintenance of ontologies [3].

In addition to this, since the main use of ontologies is making the intended meaning of a given domain available to all agents, an ontology conceptual architecture is required to represent this meaning. According to [4], there is a need to modularize the conceptual architecture to represent this intended meaning dealing with the complexity of the domain such as heterogeneous knowledge with different levels of detail of that knowledge. Therefore, the resulted designed ontologies are obtained with a high quality.

Our motivation is to build a legal ontology-based system that performs reasoning tasks in order to support the lawyers and judges in their decisions, specifically, in the criminal domain. Such a system needs a legal domain ontology for modelling the

legal norms in order to improve the efficiency of reasoning that requires the representation of the whole semantics of the criminal domain, including well-known properties such as axioms [5]. In order to reduce the complexity of the building process of the criminal domain ontology, to make it as much as possible reusable and able to perform efficient reasoning, we propose to modularize it. According to [6], a modular ontology usually contains a set of modules (component theories), implemented in same or different languages, and a set of semantic relations among those modules.
These semantic relations are obtained by applying an integration process in order to link the ontology modules together. At the end, a criminal modular ontology, named CriMOnto, is obtained. Therefore, the resulting structure of CriMOnto is a *Modular Conceptual Architecture.*

The domain application of this research is the Lebanese criminal system and the Lebanese criminal code is considered as the main textual resource since it contains the legal norms of the Lebanese criminal domain.

The main goals of this study are: survey the domain of ontology modularization for a better understanding of this domain and then provide a modular conceptual architecture for building CriMOnto as well as the modular construction process. The global view of this architecture is seen as an integration of different ontology modules developed independently.

The remainder of this paper is organized as follows: In section 2, the ontology modularization challenges are outlined. In section 3, the motivation of the study is introduced. In section 4 the ontology modularization concept and approaches are investigated. Section 5 defines the modular architecture of CriMOnto as well as the integration process. The work is evaluated in section 6. Finally, section 7 concludes the paper.

## 2 Ontology Modularization Challenges

Ontology modularization and the problem of formally characterizing a modular representation for ontologies are great challenges in the ontological engineering domain. Actually, this domain suffers from lack of the modularization theory which uncovers number of unanswered questions [7] [8]:

- How complex ontologies can be built up from parts?
- In what ways can those parts be related (mapping, integration)?
- How the structure of a modular ontology can be represented?
- to create an ontology module with a certain purpose or use-case in mind:
  - which modularity type of module is defined?
  - what properties will the resultant module exhibit?

These questions declare various aspects such as: define a modularization approach, precise the modules types and define an integration process for combining modules. By giving answers to these questions, a modular architecture for CriMOnto is introduced.

## 3 Study Motivation

According to [3], most of the existent ontologies, even if they implicitly relate several sub-domains, are not structured in a modular way. In the other hand, several works tend to combine different ontologies together, such as [9] and [10], implicitly without the explicit definition of ontology modularization concept.

For [11], in any realistic application, it is often desirable to integrate different ontologies, developed independently, into a single, reconciled ontology. This would allow for the modular design of large ontologies and would facilitate knowledge reuse tasks. Thus, there is no universal way to modularize ontologies and that the choice of a particular technique should be guided by the requirements of the considered application [3]. Therefore, an approach aims at designing a modular architecture as well as an incremental process allowing a collaborative building of CriMOnto.

The main features of CriMOnto are:

- composed of four independently developed ontology modules.
- these modules are obtained according a partial reuse of existent ontologies and semi-automatic extraction mechanisms from textual resources.
- empowered with an integration process to combine the different modules.

## 4 Ontology Modularization

The main idea of modularization originates from the general notion of modular software in the area of software engineering [12]. In software engineering domain, the modularity is a well established notion where it refers to a way of designing software in a clear, well structured way that supports maintenance and reusability [13]. However, in the ontology engineering domain, the notion of modularization and the problem of formally characterizing a modular representation for ontologies are not as well understood [14], which causes suffer in the existing work and prevents further development [3]. Despite this vagueness, ontology modularization is considered as a major topic in the field of formal ontology developments and a way to facilitate and simplify the ontology engineering process [7]. Moreover, ontology modularization has several benefits where modular representations are easier to understand, reason with, extend and reuse [13]. Therefore, using these representations tends to reduce the complexity of designing and to facilitate the ontology reasoning, development, and integration.

### 4.1 What is an Ontology Module?

Generally speaking, a module is a part of a complex system that functions independently from this system [15]. In contrast to the software engineering domain, the notion of ontology module is not clear or understood in the domain of ontological engineering [16]. There is a need to formalize and define an ontology module, particularly in terms of its requirements [17]. For [18], ontology module is considered as extractable part that can be reused outside the context of the general ontology. More clearly, an ontology module is defined by [16] as "*An ontology module is a reusable component of a larger or more complex ontology, which is self-contained but bears a* *definite relationship to other ontology modules including the original ontology*". This definition implies that ontology modules can be reused either as they are, or by extending them with new concepts, and relationships. Each ontology module is considered as ontology itself since it can be extended with new concepts and relationships. Thereby, ontology modules are themselves ontologies [12].

### 4.2 Ontology Module Criteria

The criteria of ontology modules generally aim at characterizing modular ontologies in order to evaluate the quality of modules [4]. Generally, inspired by the software engineering domain, three main criteria a module should fulfill: *self-contained*, *loose coupling* and *high cohesion* [19] [20]. Therefore, in the ontological engineering domain, some studies, such as [21], [22] and [23], believe that modularization criteria should be defined in terms of the applications for which the modules are created. They defined some ontology module criteria such as:

- *Encapsulation*: a module can be easily exchanged for another, or internally modified, without side-effects on the application.
- *Independence*: self-containment and reusability in order to improve the scalability of reasoning mechanisms.
- *Domain coverage:* generate significant module according to the different domains or topics covered by the original ontology.

### 4.3 Ontology Module Classifications

In the literature, the authors of [24] proposed to analyze modules by taking as central the following question: modules for what?

Based on this question, various classifications of ontology modules are proposed such as [8], [25], [26], and [27]. Meanwhile, the most useful classification of ontology modules is based on their content where the ontologies are classified into five main categories [28], [29] and [30]:

- *Generic*, or *top-level*, ontologies: describe generic concepts independently of a particular domain or problem.
- *Core* ontologies: in contrast to generic ontologies that span across many fields, core ontologies describe the basic categories within a domain such as law.
- *Domain* ontologies: specialize a subset of generic ontologies in a domain or subdomain, e.g., criminal law.
- *Task* ontologies: describe generic tasks or activities.
- *Application* or *domain-specific* ontologies: developed for a specific application.

### 4.4 Approaches of Ontology Modularization

Generally, modularization denotes the possibility to perceive a large knowledge repository as a set of modules, i.e. smaller repositories that, in some way, are parts of and compose the whole knowledge [31]. Therefore, an ontological modularization process is seen as a call for organizing ontologies into modules which could then be reused and combined in novel ways [7].

In the literature, three different approaches of ontology modularization are found [17] [12] [3]: ontology composition, ontology decomposition and module extraction. In this work, the ontology composition approach is used. Ontology composition aims to develop independently a set of ontology modules and assemble them coherently and uniformly, by means of integrating and mapping, to form a wider ontology. Examples of ontology composition approaches are, among others, [32], [33], [34] and [35].

## 5 Application of Ontology Modularization in the Legal Domain: CriMOnto

CriMOnto is a criminal modular domain ontology for modelling the norms of the Lebanese criminal system as a domain application for this study. In previous work [36], a middle-out approach is proposed for building CriMOnto where the ontology modularization techniques are discussed implicitly. In this work, the approach is enhanced and explicit modularization techniques are applied for this purpose. The aim of this approach is to show how ontology modularization can simplify and reduce the complexity of ontology building processes. Therefore, a modular architecture of the ontology is outlined by identifying the main modules, their number, type and criteria, as well as the knowledge to be represented in each module.

Moreover, ontology reuse process, which is now one of the important research issues in the ontology field [37], is recommended as a key factor to develop cost effective and high quality ontologies. Actually, ontology reuse reduces the cost and the time required for building ontologies from scratch [38], [39] [40]. Moreover, by reusing validated ontology components, the quality of the newly implemented ontologies is increased. According to [41], there are two main reuse processes: *merge* and *integration*. Merge is the process of building an ontology in one subject reusing two or more different ontologies on that subject. Meanwhile, integration is the process of building an ontology in one subject reusing one or more ontologies in different subjects that are maybe related. In the current work, the proposed conceptual architecture of CriMOnto is grounded on four different level concepts:

- Upper ontology module (UOM).
- Core ontology module (COM).
- Domain ontology module (DOM).
- Domain-specific ontology module (DSOM).

The proposed approach is defined by developing the modules independently by using top-down and bottom-up strategies and then combining them together to compose the whole CriMOnto (see Fig.1) [36]. From this perspective, the different modules are in different subjects since they are in different conceptual levels. Therefore, an integration process is performed to combine them.

Inspired by the integration methodology of [37], there are list of activities that precede the integration of modules into the resulting ontology such as identify the knowledge to be represented in the different modules as well as the candidate ontologies to be reused. Therefore, an analyzing and selection process will take place in order to define the existent ontologies to be reused. In the following, these activities are outlined.

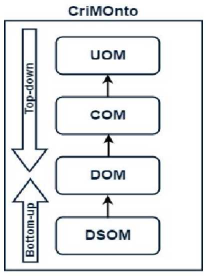


**Fig.1.** Middle-out approach for building CriMOnto.

### 5.1 Upper Ontology Module

The UOM consists of abstract concepts and relations which are effectively independent of any specific domain. For a well founded building of this module, a partial reuse of existent foundational, or top-level, ontologies can help. These ontologies are theoretically well-founded domain independent systems of categories that have been successfully used to improve the quality of conceptual models and semantic interoperability [42]. In addition to this, partial reuse of foundational ontologies can facilitate and speeding up the ontology development process by preventing to reinvent the wheel concerning basic categories [43]. In the literature, several works seek for reusing concepts from foundational ontologies in order to support in maintaining a well structured construction of domain ontologies that could serve as a future reusable artifact [44].Various foundational ontologies exist for reuse such as DOLCE [45] and UFO [46]. In this context, a selection for an appropriate foundational ontology is a crucial and difficult step since it depends on different elements such as: the purpose of building the ontology and the applicability domain. Therefore, after studying the main concepts of these foundational ontologies, UFO is the most convenient.

The unified foundational ontology UFO is a foundational ontology initially proposed by Guizzardi and Wagner [47]. UFO is developed to support the activities of both conceptual and organizational modeling [48]. Therefore, UFO permits the building of an ontology reusing some generic concepts such as *category*, *kind*, *subkind*, *relator*, *role* and *role mixin* where the ontologist does not need to rebuild these concepts. UFO is divided into three layered sets: (1) UFO-A, ontology of objects, defines terms related to endurants such as *universal*, *relator*, *role*, *intrinsic moment*; (2) UFO-B, ontology of events, defines terms related to perdurants such as *event*, *state*, *atomic event*, *complex event*; (3) UFO-C defines terms related to intentional and social entities including linguistic aspects such as *social agent*, *social object*, *social role* and *normative description* [49]. In the current work, UFO-B and UFO-C are needed to ground the criminal domain ontology for building UOM since they define some basic concepts for the criminal domain such as *Agent*, *Intentional_Moment*, *Action*, *Event*, and *Normative_Description*.

In order to make possible the activity of conceptual modeling via UFO, a conceptual modeling language, named OntoUML [50], was proposed. OntoUML uses the ontological constraints of UFO as modeling primitives and is specified above the UML 2.0 meta-model [51]. To build, evaluate and implements OntoUML models, a model-based environment is needed such as the standalone tool OLED (OntoUML Lightweight Editor) [52] or the extension of UML production-grade tool Enterprise Architect [53].

**Reusing concepts from UFO-C.** There are list of essential concepts in UFO-C to reuse for building the upper module, mainly those related to social entities such as *Agents* and *Objects* [54] (see Fig.2). *Agents* can be physical (e.g. *Person*) or social (e.g. *Organization*). *Objects* are also categorized in physical (e.g. book) and social objects (e.g. normative description). *Normative_Description* defines one or more rules/norms recognized by at least one *Social_Agent*. Regulations and constitutions are examples of normative description. In Fig.2, a fragment of the conceptual model of the upper module, which is represented in OntoUML language, is depicted.

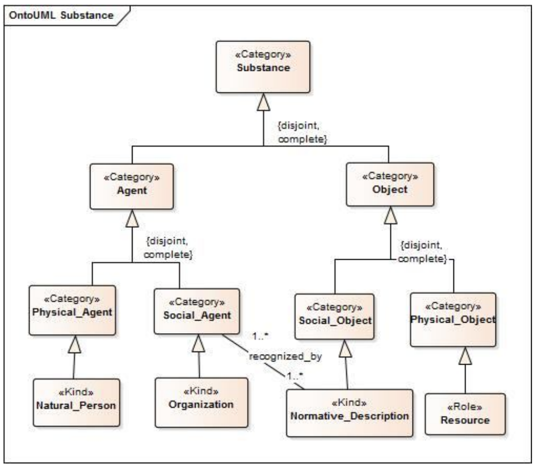


**Fig.2.** Fragment of the upper module in OntoUML.

Therefore, for knowledge representation and reasoning capabilities in the semantic web, there is a need to transform the conceptual model of the upper module represented in OntoUML language to a computational ontology language such as the Ontology Web Language (OWL). For this purpose, OLED features defined automatic transformations of the OntoUML models to OWL files [55] that can be managed using ontology editors such as Protégé (see Fig.3).

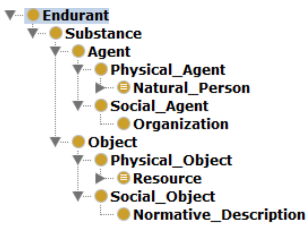


**Fig.3.** Excerpt of upper module in Protégé.

**Reusing concepts from UFO-B.** The ontology of perdurants UFO-B defines *Event*, which is a basic concept in the criminal domain (e.g. crime is an event), as a main category [54]. In UFO-B, events can be atomic or complex depending on their mereological structure [56]. Complex events are aggregations of at least two events that can themselves be atomic or complex (see Fig.4).

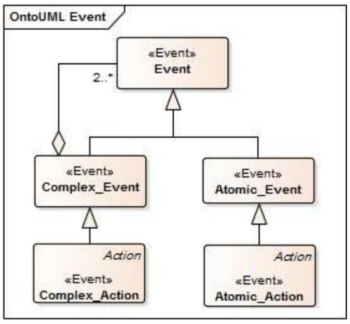


**Fig.4.** Fragment of the upper module in OntoUML.

In UFO-B, an event can be an *Action* or *Participation* (Fig.5). Actions are performed by agents and considered as intentional events caused by intentions [54]. Participation can be for agents and objects. Therefore, participation of an agent can be intentional or unintentional. Intentional participations are actions and termed here *Action_Contribution*. In Fig.5, a part of the upper ontology module represented in OntoUML is represented and Fig.6 depicts the OWL version of it.

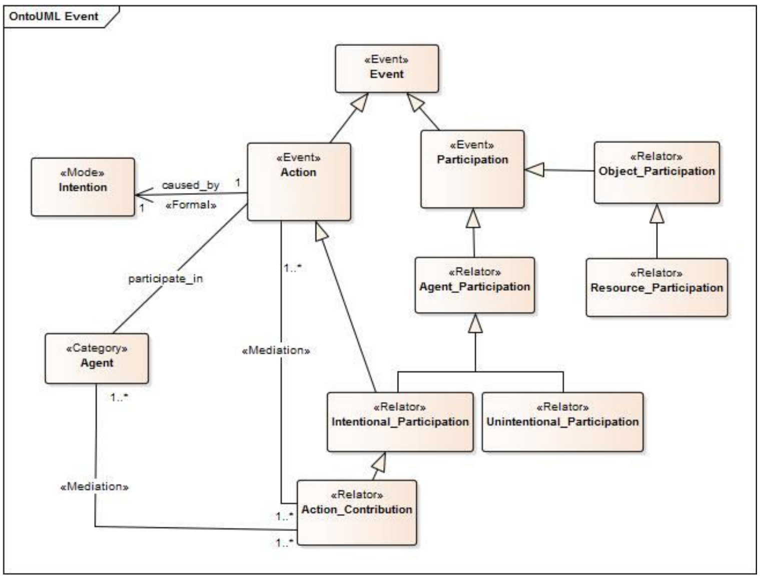


**Fig.5.** Fragment of the upper module in OntoUML.

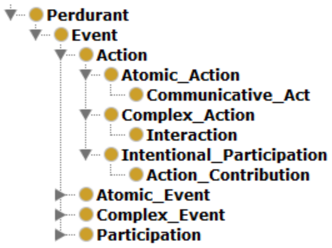


**Fig.6.** Excerpt of upper module in Protégé**.**

## 5.2 Core Ontology Module

The COM consists of concepts and relations that are common across the domains of law and can provide the basis for specialization into domain and domain-specific concepts. The same perspective can be applied, as for upper module, for reusing existent legal core ontology to build this module. Thus, it is not easy to define appropriate legal core ontology among the existent (LKIF-Core, LRI-Core, and FOLAW). Actually, LKIF-Core [57] is the most recent legal-core ontology and contains essential legal concepts such as *Medium*, *Document*, *Legal_Source*, *Legal_Document*, and *Code* (see Fig.7).

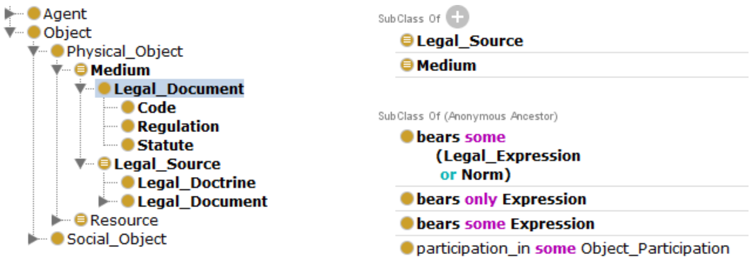


**Fig.7.** Excerpt of core concepts reused from LKIF-Core.

## 5.3 Domain Ontology Module

The DOM is composed of categories that are related mainly to the criminal domain in general such as *Criminal_Act*, *Penalty*, *Misdemeanor*, *Violation*, etc. In order to build this module, two main strategies are applied: (1) specialize the concepts and relations of the core module; (2) extract the knowledge from textual resources. The strategy of specializing the concepts and relations of the core module is applied. Therefore, concepts of the core module such as: *Creation*, *Public_Act*, *Act_Of_Law*, and *Reaction* are specialized in domain concepts such as: *Criminal_Act*, *Punishments* and *Defence* (see Fig.8 (a)). For the strategy of extracting the knowledge from textual resources, a bottom-up approach is applied as an ontology learning process, with the support of NLP techniques, for extracting the main elements of the domain ontology module from the English version of the Lebanese criminal code ( for more details refer to [58]). Examples of extracted concepts are shown in the figure such as *Defender*, *Offender*, *Accessory*, *Accomplice*, *Instigator*, *Perpetrator* and *victim* (see Fig.8 (b)).

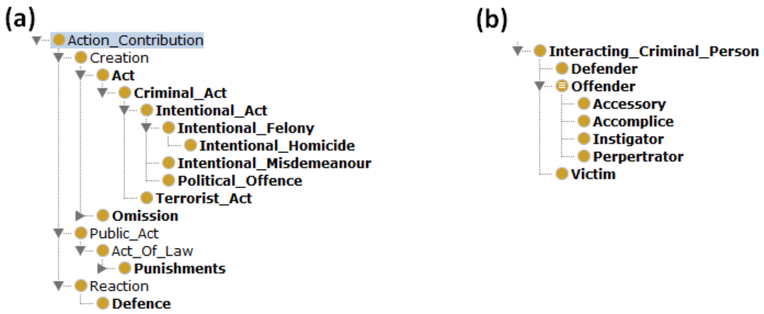


**Fig.8.** Excerpt of the domain concepts in Protégé.

### 5.4 Domain-Specific Ontology Module

The DSOM consists of concepts and relations of a specific subject domain such as the Lebanese criminal system. Domain-specific ontologies are useful in systems involved with reasoning. Thus, they should be at higher level of expressivity, in other words, rich in axioms. To build this module, since it is related directly to the Lebanese criminal system, an ontology learning process is applied on the criminal code in order to extract semi-automatically the domain-specific ontology and part of the domain ontology module [58]. The bottom-up strategy helped to generate semi-automatically an OWL ontology including the basic elements: concepts, instances, taxonomies, relations and axioms. Unfortunately, some of the generated results were inexpressive and thus insufficient for practical use. The resulted ontology module is considered as lightweight or semi-formal. Meanwhile, a more expressive ontology is required. For this reason and inspired by the work of [59], a reengineering process is applied to correct, prune and enrich the extracted ontology and make it more expressive by transforming it to heavyweight or axiom-based ontology. In the Fig.9, a fragment of the domain-specific module concepts are depicted such as *Fine_Contravention_Penalty* and *Imprisonment_Contravention_penalty*, which are specific concepts in the Lebanese criminal code, are considered as sub-concepts of the *Contravention_Penalty* concept of the domain module.

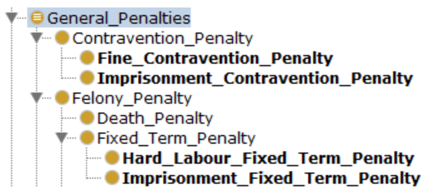


**Fig.9.** Excerpt of the domain-specific ontology module represented in Protégé.

### 5.5 Evaluation and Integration of Ontology Modules

After building the ontology modules (upper, core, domain and domain-specific), there is a need to evaluate them according to the modules criteria in order to compose the resulting ontology CriMOnto. Actually, the independent building of the modules makes them self-contained and enhances the encapsulation where any changes that affect a module will not affect the others. Moreover, the independency leads to make them reusable as well. Additionally, each module covers an adequate level of the given domain where their integration will create a complete coverage of the Lebanese criminal system.

Furthermore, an integration process will be applied to aggregate, combine, and assemble the modules. Subsequently, in the resulting ontology, regions that were taken from the integrated modules can be identified. Moreover, it is essential to consider heterogeneity resolution and related ontology matching or mapping strategies to be an internal part of ontology integration [60]. In this context, list of semantic mappings will be created among concepts of the different modules. Generally, the *mapping* concept is defined by [61] as a morphism. Meanwhile, in this study a more loosely definition is used based on some works in the literature such as [62] and claims such as “simple mappings methods are sufficient and outperform more complex methods” [63]. For ontology mappings, several studies, such as [64] and [65], have proposed a number of specialized semantics. Meanwhile, for other studies, such as [66], ontology mappings are represented as OWL2 axioms of the form *subClassOf*, *EquivalentClass* and *DisjointClass* [67]. In CriMOnto, the modules are located on vertical conceptual levels from general (upper module) to specific (domain-specific module). For this reason, the mappings will be based mainly on a parent-child, or subsumption, hierarchical relationship [68] and established manually as structural axiom of the form *subClassOf*. Thus, the hierarchical relationship is established among the concepts of modules. For this purpose, a linguistic-based matcher, such as WordNet, is used to deal with ontology mapping for calculating the similarity values between concepts [69]. Then a domain expert, knowledgeable about the semantics of legal concepts, validates the proposed mappings. Given two concepts $C_i$ and $C_j$ from the modules UOM and COM respectively, if $C_j$ is considered as a subclass of $C_i$, then the *subClassOf* axiom is added between the two concepts in the resulting ontology. In CriMOnto, *Event* is a general concept from the upper module UOM and *Legal_Event* is a legal concept from the core module COM. *Legal_Event* is considered as a subclass of *Event*. Thus, the *subClassOf* axiom is added between them (see Fig. 10).

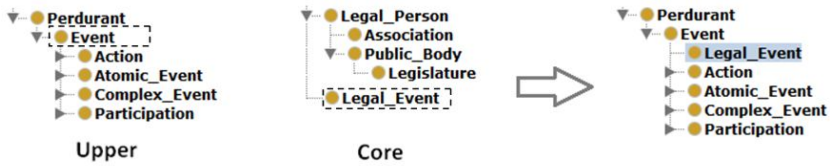


**Fig.10.** Hierarchical mapping in Protégé.

## 6 Evaluation

After building CriMOnto, there is a need to evaluate the characteristics and the validity of the resulting ontology. According to [70], evaluation is required during the whole life-cycle of an ontology in order to guarantee that what is built meets the requirements. Generally, evaluation methods consist of two parts: *verification* that ensure the ontology is constructed correctly, and *validation* that ensure the ontology represents the real world [71]. Concerning the *verification*, Ontology taxonomy evaluation method is applied [72]. This method is used manually, while integrating the four modules, with ontology checking by domain experts according to three main factors: inconsistency, incompleteness and redundancy. Meanwhile, for the *validation*, the application ontology development method is applied [72]. This method consists of building application ontology where concrete instances from real world are included in the ontology. This strategy is already done by integrating the *domain-specific* ontology module, which is the application ontology, into CriMOnto. In the *domain-specific* module, new subclasses are added that are related to the specificity of the Lebanese criminal domain. Therefore, the results are evaluated by domain experts. Actually, the presented evaluation methods are performed manually. This does not represent an obstacle since CriMOnto is not considered as a large ontology.

## 7 Conclusion

In this work, a modular approach is described to build a criminal domain ontology (CriMOnto). In the literature, several works applied implicitly the ontology modularization such as [73] for the automation domain, [74] for ill-defined domains and [75] for the enterprise architecture domain. To the best of our knowledge, there is no works in the legal domain that deal explicitly with the ontology modularization for building legal domain ontologies. For this purpose this direction is tracked. The aim of this study is to prove how ontology modularization can simplify the development of reusable legal domain ontologies with the support of ontology reuse and integration. Furthermore, a modular architecture of CriMOnto is outlined. The architecture is composed of four independent ontology modules. The upper and core modules are developed by reusing existent foundational and legal core ontologies. Meanwhile, the domain and domain-specific are constructed with the support of ontology learning process from legal texts. Furthermore, an integration process, based on simple hierarchical mappings, is applied to combine the modules in order to compose the whole CriMOnto. Finally, this approach is considered as useful track for the ontologists who seek to build domain ontologies based on existent valid ontologies and not totally from scratch.


## Acknowledgements

This work has been supported by the CLASSE project (European Union with the European Regional Development Fund ERDF), the National Council for Scientific Research in Lebanon (CNRS) and Lebanese University.


## References


1. Corcho, O., Fernández-López, M., and Gómez-Pérez, A.: Ontological Engineering: What are Ontologies and How Can We Build Them? In C. Jorge (Ed.), Semantic Web Services: Theory, Tools and Applications (pp. 44-70). Hershey, PA, USA: IGI Global. (2007).

2. Pathak, J., Johnson, T. M., & Chute, C. G.: Modular ontology techniques and their applications in the biomedical domain. Integrated Computer-Aided Engineering, 16(3), 225–242 (2009).
3. d'Aquin, M., Schlicht, A., Stuckenschmidt, H., and Sabou, M.: Ontology modularization for knowledge selection: Experiments and evaluations. In R. Wagner, N. Revell, and G. Pernul, editors, 18th International Conference on Database and Expert Systems Applications (DEXA'07), volume 4653 of LNCS, Springer, pages 874–883, (2007).
4. Gangemi, A., Catenacci, C., Battaglia, M.: Inflammation ontology design pattern: an exercise in building a core biomedical ontology with descriptions and situations. Stud Health Technol Inform 102:64–80, (2004).
5. Fürst, F., and Trichet, F.: Integrating Domain Ontologies into Knowledge-Based Systems. In FLAIRS Conference (pp. 826-827). and Physical Layer (PHY) Specification, IEEE Std. 802.11, (2005).
6. Wang, Y., Bao, J., Haase, P., and Qi, G.: Evaluating Formalisms for Modular Ontologies in Distributed Information Systems. In 1st International Conference on Web Reasoning and Rule Systems, pages 178–193. LNCS 4524, Springer, (2007).
7. Hois, J., Bhatt, M., and Kutz, O., Modular Ontologies for Architectural Design. In Proc. of FOMI-09, volume 198 of Frontiers in Artificial Intelligence and Applications, Vicenza, Italy, IOS Press, (2009).
8. Khan, Z. C. and Keet, C. M.: Toward a framework for ontology modularity. In: Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists, SAICSIT '15, Article No. 24, (2015).
9. Francesconi, E., Montemagni, S., Peters, W., and Tiscornia, D.: Integrating a bottom-up and top-down methodology for building semantic resources for the multilingual legal domain. In: Proc. Legal Texts. Where the Language of Law Meets the Law of Language, LNAI 6036, Springer, Berlin, pp. 95-121, (2010).
10. Saias, J. and Quaresma, P.: A methodology to create legal ontologies in a logic programming based web information retrieval system. In: Proc. Law and the Semantic Web Legal Ontologies, Methodologies, Legal Information Retrieval and Applications, Springer, Berlin, pp. 185-200, (2004).
11. Turlapati, V. and Puligundla, S.: Efficient module extraction for large ontologies. In Knowledge Engineering and the Semantic Web, volume 394 of Communications in Computer and Information Science, pages 162–176. Springer Berlin Heidelberg, (2013).
12. Abbes, S. B., Scheuermann, A., Meilender, T., and d'Aquin, M.: Characterizing modular ontologies. In: 7th International Conference on Formal Ontologies in Information Systems (FOIS), pp. 13–25, (2012).
13. Grau, B. C., Horrocks, I., Kazakov, Y. and Sattler, U.: A logical framework for modularity of ontologies. In: Proceedings of IJCAI'07, pages 298–303. AAAI Press, (2007).
14. Grau, B. C., & Kutz, O.: Modular ontology languages revisited. In: Proc. of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition, (2007).
15. Konev, B.; Lutz, C.; Walther, D.; and Wolter, F.: Formal properties of modularization. In Stuckenschmidt, H.; Spacciapietra, S.; and Parent, C., eds., Ontology modularization, volume 5445 of LNCS. Springer. 25–66, (2009).
16. Doran, P.: Ontology reuse via ontology modularization. In: Proceedings of Knowledge-Web PhD Symposium, pp. 1-6, (2006).
17. Bezerra, C., Freitas, F., Zimmermann, A., and Euzenat, J.: ModOnto: A tool for modularizing ontologies. In: Proc. WONTO-08, volume 427 of ceur-ws. org , (2008).
18. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and web ontologies. In: KR. pp. 198–209 (2006).

19. Stuckenschmidt, H. and Klein, M.: Reasoning and change management in modular ontologies. Data Knowledge Eng., 63(2):200–223, (2007).
20. Stuckenschmidt, H, Klein M.: Integrity and Change in Modular Ontologies. 18th International Joint Conference on Artificial Intelligence, pp. 900–905, (2003).
21. d'Aquin, M., Schlicht, A., Stuckenschmidt, H., and Sabou, M.: Criteria and evaluation for ontology modularization techniques. In: H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors, Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, volume 5445 of LNCS, pages 67–89. Springer, (2009).
22. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study*. Technical report, University of Manchester, Available from http://www.cs.man.ac.uk/%7Eschneidt/publ/modstrucreport.pdf .
23. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study. In V. Haarslev, D. Toman, and G. Weddell (eds.) Proc. of DL 2010. ceur-ws.org, vol. 573. (2010).
24. Borgo, S., Goals of modularity: A voice from the foundational viewpoint. In O. Kutz and T. Schneider, editors, Fifth International Workshop on Modular Ontologies (WOMO'11), volume 230 of Frontiers in Artificial Intelligence and Applications, pages 1–6. IOS Press, (2011).
25. Studer, T.: Privacy preserving modules for ontologies. In A. Pnueli, I. Virbitskaite, and A. Voronkov, editors, PSI'09, volume 5947 of LNCS, pages 380–387. Springer, (2010).
26. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The Modular Structure of an Ontology: Atomic Decomposition and Module Count.
27. Grau, B. C., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. In: Proc. of ISWC/ASWC-07. Volume 4825 of LNCS,183–196, (2007).
28. Guarino, N., Carrara, M. Giaretta, P.: An Ontology of Meta-Level Categories. In J Doyle, E Sandewall and P Torasso (eds.), Principles of Knowledge Representation and Reasoning: Proceedings of KR94. San Mateo, CA, Morgan Kaufmannn, (1994).
29. Guarino, N.: Understanding, Building, And Using Ontologies. International Journal of Human-Computer Studies, Volume 46 Issue 2-3, 293 – 310, (1997).
30. Van Heijst, G., Schreiber, A. Th., Wielinga B.G.: Using Explicit Ontologies in KBS Development. Int. Journal of Human-Computer Studies, (1997).
31. Stuckenschmidt H., Christine P. and Spaccapietra S.: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. Springer Lecture Notes in Computer Science, (2009).
32. Ben Mustapha, N., Baazaoui-Zghal, H., Moreno, A., Ben Ghezala, H.: A dynamic composition of ontology modules approach: application to web query reformulation, International Journal of Metadata, Semantics and Ontologies, Vol. 8, Issue 4, pp. 309-321, (2013).
33. Bezerra, C., Freitas, F., Euzenat, J., Zimmermann, A.: An Approach for Ontology Modularization, (2008).
34. Dmitrieva, J., Verbeek, F. J.: Creating a new Ontology: a Modular Approach, Article,( 2010).
35. Steve, G., Gangemi, A., Pisanelli, D.: Integrating medical terminologies with onions methodology, (1998).
36. El Ghosh, M., Naja, H., Abdulrab, H.and Khalil, M.: Towards a Middle-out Approach for Building Legal Domain Reference Ontology. International Journal of Knowledge Engineering, vol. 2, no. 3, pp. 109-114, September, (2016).
37. Pinto, H. and Martins, J.: Ontology Integration: How to Perform the Process. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 71-80, (2001).

38. Bontas, E. P., Mochol, M., and Tolksdorf, R.: Case studies on ontology reuse. In: Proceedings of the IKNOW05 International Conference on Knowledge Management, 74, (2005).
39. Caldarola, E. G., Picariello, A., and Rinaldim A. M.: An approach to ontology integration for ontology reuse in knowledge based digital ecosystems. In: Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems, pages 1– 8. ACM, (2015).
40. Modoni, G., Caldarola, E., Terkaj, W. and Sacco, M.: The knowledge reuse in an industrial scenario: A case study. In eKNOW 2015, The Seventh International Conference on Information, Process, and Knowledge Management, pages 66–71, (2015).
41. Pinto, S. H., Gomez-Perez, A., Martins, J. P.: Some Issues on Ontology Integration. In Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, (1999).
42. Guizzardi, G.: The role of foundational ontology for conceptual modeling and domain ontology representation. In: Proc. 7th International Baltic Conference on Databases and Information Systems, pp. 17-25, (2006).
43. Keet, M.: The use of foundational ontologies in ontology development: An empirical assessment. In: Proc. 8th Extended Semantic Web Conference, Greece, vol. 6643, pp. 321-335, (2011).
44. Rosa, D.E., Carbonera, J.L., Torres, G.M., Abel, M.: Using events from UFO-B in an ontology collaborative construction environment. CEUR-WSX **938**, 278–283 (2012).
45. Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A., Wonderweb Deliverable D18 (ver. 1.0), Ontology Library, (2003).
46. Guizzardi, G. and Wagner, G.: A unified foundational ontology and some applications of it in business modeling. CAiSE Workshops, vol. 3, pp. 129-143, (2004).
47. Guizzardi G. and Wagner G.: Using UFO as a Foundation for General Conceptual Modeling Languages. In Theory and Application of Ontologies, Springer, (2010).
48. Melo, S., Almeida, M. B.: Applying Foundational Ontologies in Conceptual Modeling: A Case Study in a Brazilian Public Company. (2014).
49. Guizzardi, G., Wagner, G.: Towards ontological foundations for agent modelling concepts using the unified foundational ontology (UFO). Agent-Oriented Inf. Syst. II(3508), 110–124 (2005).
50. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Phd Thesis. Enschede, The Netherlands: Telematica Instituut, (2005).
51. Guerson, J., Sales, T. P., Guizzardi, G., & Almeida, J. P. A.: OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies. In Enterprise Distributed Object Computing Workshop (EDOCW), IEEE 19th International (pp. 144-147). IEEE. (2015).
52. http://code.google.com/p/ontouml-lightweight-editor/ , last accesses 19/4/2017
53. http://www.sparxsystems.com/products/ea/, last accessed 19/4/2017
54. Guizzardi, G., Falbo, R. A., and Guizzardi, R. S. S. : Grounding software domain ontologies in the unified foundational ontology (UFO): The case of the ODE software process ontology. In: Proceedings of the Iberoamerican Workshop on Requirements Engineering and Software Environments,. pp. 244–251, (2008).
55. Barcelos, P.P.F., dos Santos, V.A., Silva, F.B., Monteiro, M.E., Garcia, A.S.: An automated transformation from OntoUML to OWL and SWRL. In: ONTOBRAS 2013. CEUR Workshop Proceedings, vol. 1041, pp. 130–141. CEUR-WS.org (2013).
56. Guizzardi, G., Wagner, G., Falbo, A., Guizzardi, R. S. S., and Almeida, J. P. A.: Towards Ontological Foundations for the Conceptual Modeling of Events. In: 32th International Conference, ER 2013, pp. 327–341, (2013).

57. Hoekstra, R., Breuker, J., Bello, M., D. and Boer, A.: The lkif core ontology of basic legal concepts. In: Proc. the Workshop on Legal Ontologies and Artificial Intelligence Techniques, CEUR Workshop Proceedings, vol. 321, pp. 43-63, (2007).
58. El Ghosh, M., Naja, H., Abdulrab, H. and Khalil, M. : Ontology learning process as a bottom-up strategy for building domain-specific ontology from legal texts. In: Proceedings of the 9th International Conference on Agents and Artificial Intelligence – vol 2- ICAART, ISBN 978-989-758-220-2, 473-480, (2017).
59. Gómez-Pérez, A., Rojas-Amaya, D.: Ontological Reengineering for Reuse. In: proc. EKAW'99, of the 11th European Workshop on Knowledge Acquisition, Modeling and Management, pages 139-156, Springer, (1999).
60. Caldarola, E. G., Picariello, A., and Rinaldi, A. M.: An approach to ontology integration for ontology reuse in knowledge based digital ecosystems. In: Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems, pages 1– 8. ACM, (2015).
61. Kalfoglou, Y., Schorlemmer, W.M.: Ontology mapping: The state of the art. In: Semantic Interoperability and Integration (2005).
62. Julia Dmitrieva and Fons Verbeek: Modular Approach for a new Ontology, Proceedings of 5th International Workshop on Modular Ontologies WoMO, (2011).
63. Ghazvinian, A., Noy, N.F., Musen, M.A.: Creating mappings for ontologies in biomedicine: Simple methods work. In: AMIA 2009 Symposium Proceedings (2009).
64. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proc. of IJCAI, 348-353, (2007).
65. Borgida, A., Sera_ni, L.: Distributed description logics: Assimilating information from peer sources. J. Data Semantics 1, 153-184, (2003).
66. Jimenez-Ruiz, E., Cuenca Grau, B., Horrocks, I., and Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences (technical report). Technical report, Universitat Jaume, University of Oxford, (2008).
67. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. J. Web Semantics 6(4), 309-322, (2008).
68. Wang, Y., Liu, W., Bell, D.: A Concept Hierarchy Based Ontology Mapping Approach. In: KSEM. pp. 101–113 (2010).
69. Miller G. A.: WordNet: A lexical database for english. Communications of the ACM 38, 39-41, (1995).
70. Hartmann, J., Spyns, P., Giboin, A., Maynard, D., Cuel, R., Suárez-Figueroa, M. C., Sure, Y.: Methods for ontology evaluation, *Knowledge Web Deliverable D1.2.3* (2004)
71. Gómez-Pérez, A., Fernandez-Lopez, A., Corcho, O. : Ontological Enginering, Springer-Verlag, London, UK, (2004).
72. Gómez-Pérez, A: Evaluation of Ontologies, International Journal of Intelligent Systems, 16, Chichester, UK, pp. 391-409, (2001).
73. Legat, C.: Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain, Conference Paper, World Congress, (2014).
74. Thakker, D., Dimitrova, V., Lau, L., Denaux, R., Karanasios, S., and YangTurner, F.: A priori ontology modularisation in ill-defined domains. In: Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11, 167–170, USA, (2011).
75. Bakhshandeh, M., Antunes, G., Mayer, R., J. Borbinha and A., Caetano: A Modular Ontology for the Enterprise Architecture Domain. In: Proceedings of the 2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOCW '13, Pages 5-12, (2013).