AN ANALYSIS OF COMMIT-TIME REQUIREMENTS FOR ONTOLOGY VIEWS

Mohammad Nazir Ahmad[1], Robert M. Colomb[2], Shazia Sadiq[3]

1,3 School of IT and Electrical Engineering, the University of Queensland, Australia
{nazir[1], shazia[3]}@itee.uq.edu.au
[2]Faculty of Computer Science and Information Systems, the University Technology of Malaysia, Malaysia
colomb@utm.my

Abstract: One of the major uses of the ontology is to support interoperation of information systems which that ontology is a result of Interlocking Institutional Worlds (IWs). In this context, typically, application development in collaborative setting requires users to commit particular applications to an ontology. In such environment, the user employs an ontology server to identify ontology objects the user considers essential to commit the application. Since there are many users, therefore their different perspectives will require server to generate views on the ontologies. The ontology views are fairly established in ontology modularization research but in the context of IWs we argue that the problems may be distinct. Therefore, this paper is to define several ontology views problems and subsequently propose some sort of requirements for ontology views with respect to IWs. Furthermore, we discuss our analysis in the sense of ontology modularization approaches and argue that our problems make differences thus require another new approach to solve it.

Keywords: Ontology, Institutional World, Interlocking Institutional Worlds, Ontology Server, Semantic Web.

1. INTRODUCTION

An ontology is a specification of a conceptualization [1]. A large institution may have hundreds information systems and in turn may interoperate with other institutions in a global economy. In this context, ontology is prominently used to support information systems interoperability where one of the main important aspects will involve application development in collaborative setting. In this situation requires users to *commit* particular applications to an ontology. Such environment, the user employs an *ontology server* to identify ontology objects the user considers essential to commit the application. In this purpose, the server usage is for commit-time phase [2-4]. Due to many users are involved, therefore their different *perspectives* will require *views* on the ontologies in order to have a *portion* of a *large* and *complex* ontology. The views represent parts of the ontology which is a *specification.* We argue that the process of generating views is supposed to be propagated on what occurs in its *conceptualization*. For example, any server operations such as subscribe to views must be triggered by its conceptualization. In fact, in interoperating systems, pretty well all players are *active*. Unfortunately, current research [5-19] regards ontology view is only its specification and do not elaborate what *"sort of things"* there are in the conceptualization. Therefore, the generated view would lead to *intuitively* incorrect and incomplete.

The rest of the paper is structured as follows: Section 2 provides basic concepts of Institutional World (**IW**) and Interlocking Institutional Worlds (**IWs**). Section 3 presents our point of view on ontology views, while section 4 we define several important problems for ontology views with regard to the context of **IWs**. To address highlighted problems, section 5 subsequently defines some important requirements for ontology views, whereas in section 6 we discuss our analysis from section 4 and 5 respectively in the context of current approaches to ontology views. Section 7 concludes some remarks and outlines our future works.

2. INSTITUTIONAL WORLD AND INTERLOCKING INSTITUTIONAL WORLDS

The concept of institutional world and interlocking institutional worlds are defined in [20]. The concept is grounded by a credible theory of institutional facts [21] which arguing that there are two different facts; *brute fact* and *institutional fact*. The brute fact is independent on human society or *context* while institutional fact is not. Our world is full of institutional facts. One's name is an institutional fact. Being given a name is an *act*, called a *speech act*, performed by one's parents and government department acting in cooperation. The speech act is recorded in some way, such as on a birth certificate or passport. The record of the speech act is actually an institutional fact. Likewise, information systems are almost exclusively concerned with storing institutional facts. Most messages between information systems are speech acts. The fact that someone is a customer (stored in the *Customer* table) is an institutional fact. The customer's name is an institutional fact (created in a speech act by the person's parents). The customer's credit rating is an institutional fact created in a speech act by the company's accounting department. The information systems' business rules enforce the context rules determining the validity of the speech acts, and the systems themselves keep track of how the world changes as a result.

Once an interoperating community is established, it can generate a large number of institutional facts. This collection of integrated speech acts and consequent institutional facts made by a particular institution as that institution's *institutional world*. This institutional world is basically the *conceptualization* of which the ontology is the *specification*. Even though institutions are generally more or less autonomous, they do sometimes cooperate with each other, so that their institutional worlds will interlock thus interlocking ontologies. If we think of there are many institutions interoperate thus may generate a large number of institutional facts. We can therefore think of the *overall* conceptualization is a result of **IWs,** forming a larger **IW** and consequently large and complex ontology (e.g., Olympic ontology).

There are several ways two institutional worlds can interlock (the following list is not intended to be exhaustive): (i) The speech act creating them is *performed by both* institutions (e.g., a purchase involves a buying speech act by one institution and a selling speech act by another); (ii) An institutional fact in one institution is *part of* the context of a speech act performed by the other (e.g., a student's speech act of enrolling in a course depends on the prior speech act of creation of the course by the University. The institutional fact of the course is part of the context of the student's enrolment speech act.); (iii) A speech act by one institution can be performed under *licensed by* another. (e.g., a taxi ride, where the taxi operates under licence of a government agency. So is the conduct of sporting events by an **OCOG**. The rules of the events are specified by the sporting federations, so the **OCOG** is in effect acting as an agent of the sporting federations.); (iv) One institution may *constrain* the speech acts of another. (e.g., the **IOC** constrains all events in the Olympics to award gold, silver and bronze medals, a constraint on the rules of the events created by the sporting federations).

It seems to us that two institutions can interoperate if they share their system of institutional facts. Refer to the ontology definition in [1], it makes sense that this sort of *"things"* reflects to the aspect of *"conceptualization"*. In order to make these *"things"* being shared amongst agents (e.g., human or applications) in a given domain, it must be *explicitly* captured in term of concrete artifacts which reflecting to the aspect of *"specification"*. Conventionally, we use conceptual representation system such as **UML ODM** [22] for domain representation. The main focus of this paper is to look at problems in ontology views. We submit here how does the concept of **IWs** contribute to the ontology views problem? To answer, we first need to define our position on ontology views.

3. ONTOLOGY VIEWS

We agree with four purposes of ontology views defined in [23]: (i) to provide a *manageable portion* of a larger ontology for the localized applications and users; (ii) enable *precise extraction* of sub-ontologies of a larger ontology that commits to the main ontology; (iii) enable *interoperability* between large ontology bases and applications; (iv) enable localized *customization* and usage of the portion of a larger ontology. However, there exists no *standard* way to ontology views [23] and most of them have been developed *independently* of the needs of a concrete application domain and some others approaches are based on its own assumptions and contexts [18]. We agree approaches to ontology views should be defined under a concrete application domain (e.g., ontology server for supporting **IWs**). In this way, they would have a proper and focus evaluation. One of important aspects for ontology views is a representation system the views basing on. We argue that views on ontology supposes to be generated from a base ontology (source ontology) represented using a standard modeling languages such as **UML ODM.** However, most of traditional modeling languages suffer from *ontological semantics* [24] and are not sufficient to represent many conceptualization problems.

We conclude here approaches to ontology views may differ from each other with regards to its own assumption and context. To have a proper evaluation, the approach needs to be developed in a concrete *application domain* such as ontology server for supporting the domain of **IWs**.. In addition, it should make contributions towards a *standard* way of ontology views methods. Next, we define several problems in ontology views tailored to our context.

4. PROBLEMS IN ONTOLOGY VIEWS IN THE CONTEXT OF IWs

This section defines ontology views problems in the context of **IWs.** The **SW** as a society of **IWs** [20]. The Olympics is a good example of an institutional world (thus an ontology e.g., Olympic ontology) formed by the interlocking of many institutional worlds because it is large, complex, well established, familiar and very well published, especially in its sporting dimension, but also in its business dimensions due to the requirement for accountability to all stakeholders. As illustrated in Fig. 1, an overall structure denotes an Olympic ontology. Each fragment is an ontology module which occurs to us is respectively part of an Olympic ontology. These fragments contributed by many institutions which may range from private and public institutions to a large number of Olympics functional groups interoperate *mutually* to make the game a success. Of course this scenario will involve a large number of applications to interoperate based on this large ontology.

Let **Ö** and **Ü** denote owner and user respectively. **Definition 1:** An owner, **Ö** is an ontology owner, who creates the world (specify ontology). **Definition 2:** A user **Ü** is an ontology user who uses a portion of the ontology. Besides these, we also need definition for **O** and **M** which denotes ontology and ontology module respectively. **Definition 3:** An ontology **O** or called as a *base ontology* or *source ontology* is a specification of a conceptualization (institutional world as a problem domain) which consists of a collection of objects such as Classes **C(O),** Properties $\mathbf{P}^{\wedge}(\mathbf{O})$, Properties Instance $\mathbf{P}^{*}(\mathbf{O})$ and Individuals **I (O).** **Definition 4:** A module

M is an ontology module or *fragment* of an ontology. It is a result of modularization process $\mathbf{M_p}$. If **M** is created by a user **Ü** then denoted as $\mathbf{M^U}$, while the ontology module **M** which is specified by an ontology owner **Ö** thus depicted as $_O\mathbf{M}$.

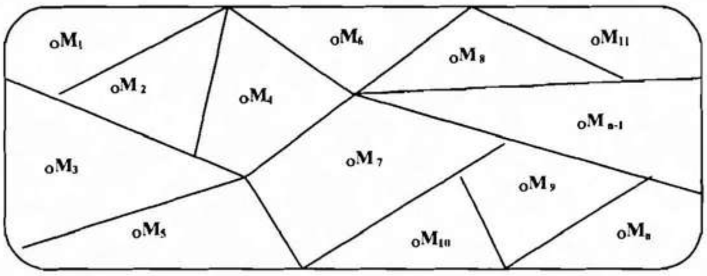

Fig. 1. The **IWs** adapted from [3]

As shown in Fig. 1, the **IWs** constitute a larger and complex ontology **O** which composed of many ontology modules $_O\mathbf{M}$ as parts. From our definitions here, we define several main problems will have to be resolved in the context of **IWs.** Those problems are defined as follows:

4.1 Problem (P1) – Changes in Institutional World

To keep in mind, an ontology **O** provides a *coordinated* system of institutional facts. Refer to **IWs** definitions in section 2, the **IWs** may happen in many ways and each institution interoperates with a *mutual understanding.* In order to commit to an ontology **O**, some institutions might need to make changes in its institutional world. For example, the way to determine a winner in match-oriented games like Badminton (defined by an owner **Ö (IBF)**) needs to be based on how **IOC** may *constrain* the speech acts of another (e.g., the **IOC** constrains all events in the Olympics to award gold, silver and bronze medals, a constraint on the rules of the events created by the sporting federations). The change of "Olympic Badminton Winner Rules" is an institutional fact created in **IBF**'s institutional world generated by so-called a *performative speech act.* Some users might be happy with a *coarse-grained* (performative) speech act (e.g., create only classes), while some others might want to have a more *fine-grained* speech acts (e.g., create classes, properties and individuals).

However, the element changes **(e.g., C(O),** $\mathbf{P^{\wedge}(O)}$, $\mathbf{P^{*}(O)}$ or **I (O))** of **IBF**'s institutional world, the other institutions (e.g., **OCOG, NOC**'s participants) need to know about it. So the generated views to other participants should be kept updated or informed with changes in underlying ontology due to changes in a particular institution's institutional world. Furthermore, besides *announcement* about changes, other institutions may make a *query* about it. Information about changes created in so-called an informative speech act (*announcement* or *query*) are institutional facts as well. We submit here, the process of views creation should be governed by interaction between these two types of speech acts: *performative* and *informative*. However, how can we specify these sorts of things?

4.2 Problem (P2) – User's Perspective

Typically, the ontology views are computed according to the user's *perspective*. Gahleither et al. [16] propose process-oriented views on ontologies. For example, a technician who designs new car engine needs information which is different from the information a worker at the assembly line or a car dealer requires for the customers. In this sense, *"process"* is a user's context in which a user works determines the user's view or called as a *"perspective"* on the available knowledge regards to that user. In **IWs**, there are users who are responsible (has a role) for a particular application to commit to the ontology. So the user's perspective here characterized by the *"application contexts (type of application, application's relevant objects, etc)"* or simply termed here *user-application* determining what sort of view could be generated for that application. For example, **GIS** application may need ontology structure on geographic information (e.g., location) while a tourism application might need it as well together with information on various tourist spots (e.g., hotels, restaurants). So the views for the user-application in the context of **GIS** and tourism application must be distinct

However, what sort of view can we define that respect to the user's perspective in the context of **IWs**?

4.3 Problem (P3) – Modeling Language of Views

Note that the ontology views are generated (e.g., extraction process) based on a source ontology **O**. Subject to a user Ü, the results of this process $\mathbf{M_p}$ is an ontology module $\mathbf{M^U}$. However, in [23] highlight two important issues in the case of ontology views; (i) Unlike database views, ontology views are not just an extracted portion of **O**, but itself is a *new* interpretation of the base ontology; (ii) The representation of meaningful of such ontology views or resulted $\mathbf{M^U}$ to the users is necessary and it supposes to easily transformed to machine or user-application readable notations. Furthermore, an extraction mechanism will need to provide *ontological consistency* of the elements in a given $\mathbf{M^U}$. For example, given

C(O) there may be other **C(O)** on which the required **C(O)** depend. Such notion of *ontological dependency* is crucial to preserve from ontological inconsistency.

Unfortunately, the notion of ontological dependence in many traditional conceptual modeling languages such as **ER** [25], **UML** [26], **OWL** [27] or **UML ODM** is simple and not sufficient. Therefore, if we base on these sorts of representation systems, the views will be produced intuitively incorrect. Though, what sort of *"system"* is appropriate to solve this problem?

4.4 Problem (P4) – The Notion of "Private" and "Public"

In **IWs**, many institutions are generally more or less autonomous although *loosely coupled.* They cooperate each other thus their institutional worlds will interlock. Each institution may have its way of doing things subject to the context they interlock. Two institutions interlock may involve some aspect of the ontology and some other aspects might be not of interests. For example, we may have an online broadcaster application. This application may interlock with almost all sporting institution. So it can make any reports about sports competition or events. In a series of *"breaking sports news"*, at one time, the broadcaster might report about badminton competition. At the other time, it may report on other sports competition. Something that is *"interests"* to a particular user's application may be not for some other users (because the application might not be interested) might best be thought of as an object that users can be *subscribed to*. So this sorts of objects we refer here as *"public"*. In contrast, the objects which are not accessible to anyone else we refer here is *"private"*. The aspect of the ontology that can only be accessed by ontology owner represents what we meant here as *"private"*. There is evidence a role for *private* in the ontology. For example is the discussion of the **SIC** in [2]. The **SIC** is to the public a system of declared subclasses, but to the **US** Bureau of Labor Statistics (**BLS**), the owner, a system of defined subclasses. But the rules for defining the subclasses, even if they were published, rely on data that is held by the **BLS** on a commercial-in-confidence basis, so is private in the **SE** sense.

The notion of *private* and *public* is well-established in **SE** (Software Engineering). However we distinguish here *"public"* is not meant that any subscribers can change the *"world"* but they can only *"see"* the *"world"* as public due to their *"interests"*. While the *private* is almost similar to **SE** since only the owner of that particular object can make any change on the *"world"* is. We submit here how can we define the views which contain only what object that user subscribes to?

4.5 Problem (P5) – The View Extraction Mechanism

This problem closely relates to the problem 3 since the extraction process is tightly depended on what sort of systems to representing the ontology **O**. A few paradigms have been adapted in many ontology modularization approaches as discussed in section 6. Some approaches do not destroy a whole ontology but some other approaches do. In **IWs,** a whole ontology **O** formed by the interlocking institutional worlds as *parts* and each of them are more or less *autonomous*. Any approach $\mathbf{M_p}$ to partitioning the **O** is not appropriate in the context of **IWs.**

An approach to *traverse* the **O** would be a good mechanism since it does not destroy the overall structure **O** where its parts belong to different kind of owners. However an extraction of views should have a kind of limitation e.g., *specific* or *generic* traversal might be involved. Nevertheless, what sort of mechanism do we need to develop to perform this task? In addition, how far this mechanism will preserve *ontological consistency* of a resulted ontology module $\mathbf{M^U}$? To address proposed problems, in the following section we define several requirements (a conceptual-stage solution) for our ontology views.

5. OUR APPROACH: THE REQUIREMENTS FOR ONTOLOGY VIEWS IN THE CONTEXT OF IWs

This section defines six requirements purposely to provide some *key* ideas (conceptual solution) for our problems. Fig. 2 illustrates our notion of ontology views.

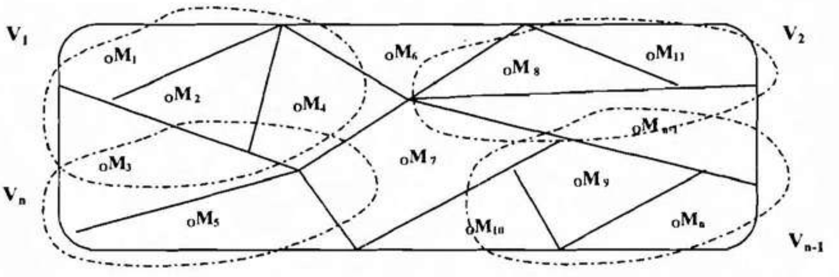

Fig. 2. The notion of ontology views in **IWs**.

A single view, $\mathbf{V_1}$ is resulted from an *extraction* process $\mathbf{M_p}$. Some similar works which in line (e.g., traversal or extraction paradigm) with our view have been proposed in [8, 11, 28].

5.1 Requirement (R1) – The Views Require a "Coordination" Mechanism

To approach **P1**, our **R1** requires a kind of *"business model"* to coordinate how users as well as owners should work for *using* or/and *specifying* a structure of interlocking ontologies (e.g., coordinate speech acts – performative and informative). To our knowledge, some famous **LAP** (Language-Action Paradigm) approaches like **DEMO** [29] would fit well to the concept of **IWs**. Such a business model, focusing on coordination is seen as foundational for the development of supporting software and in fact there is a growing interest to look forward and apply **LAP** to the future of the Semantic Web [30].

However, our intention is to contribute to the standard approach to ontology and ontology server, therefore using a *defacto* standard like **UML** is a good modeling choice. Unfortunately, there is a different paradigm between **UML** and **DEMO** which explained as *"reaction view"* versus *"action view"* in [31]. To resolve these discrepancies, to use **DEMO** as preliminary work such as [32] is a good idea and then constructing a profile for it, possibly profiling features of actions (speech acts) as well as of the **UML** classes model. This may show us how **DEMO** could contribute to the **UML**-based information systems development with particular focus on ontology and ontology server engineering. In other words, working on this direction is argued as towards a standard approach to ontology and ontology server as mainstream as **MDA** to ontology that resulting a **UML ODM** [22, 33].

5.2 Requirement (R2) – The Views and the "Perspective"

For *characterizing* a user-application (**P2**), our **R2** requires we define the notion of *"contextual fact (ç)"* and *"perspective (**þ**)"* which would be parts of our *views properties*. Since we are concerned on ontological dependency (see **P3**), our views suppose to consist of *interrelated (relevant)* ontology objects as a *whole*, **£** (e.g., interrelated classes with relationship and perhaps individuals) determined by the notion of *ontological dependency*, **OD**. This relevant aspect of the ontology (specified by owners) is relative to a starting point supplied by the user Ü. A given **£** belongs to the domain **Đ** as captured by the ontology **O**.

However, subject to a given user Ü, each object (e.g., class) in **£** is *a contextual fact*, **ç** that makes sense to that user's point of view or called *perspective*, **þ**. The same object **ç** may make sense to the different **þ**. For example, in a given $\mathbf{V}_1$, there is £_1 consists of interrelated ç_1 that make sense to a given þ_1. The same ç_1 (say ç_2) might make different sense in another given þ_2, for example. So, at this stage we would define our *conceptual* ontology view, **V** = (**þ**, **£**) where **þ** refers to *"who"* is a stakeholder (e.g., a particular user) that subscribe to a *view*, **V** of the interlocking ontologies structure, at given point in time. While, **£** is a "*module*" consists of interrelated *contextual fact,* **ç** in a given **þ** at given point in time. Now, think of Olympics. Consider interlocking between a government agency **G** and a standard body of tourism industry, **T**. Assume the "user-**GIS**" and "user-TravelSystem" might use travel ontology (defined by **T**) and geographic ontology (defined by **G**) respectively.

There are ontology objects such as "GeographicArea", "City" and "Country" which could be defined by **G.** The core tourism ontology would define concepts such as ActivityProvider to link an Activity with a ContactAddress. There could be a set of subtypes of activities such as BungeeJumping or Caving, and these could be categorized into types like AdventureActivity. On one hand, objects such as "GeographicArea", "City" and "Country" are interrelated *contextual facts* **ç** in **£** as a whole in the sense of a given perspective (**þ**) which is a viewpoint to the "user-**GIS**". On the other hand, for each following object; "ActivityProvider", "Activity", "ContactAddress" ("GeographicArea", "City", "Country"), "BungeeJumping", "Caving" is a *contextual fact* **ç** in another **£** as a whole regards to another perspective (**þ**), in this case is a viewpoint to the "user-TravelSystem". Note that, in this example we simply use common sense to determine the ontological dependence.

We need a View Constraint Specification Language (**CVSL**) for specifying our views definition and demonstrating them in a case study of **IWs**. Unfortunately, a semantic web language like **OWL** has no mechanism to define constraints on views as well as general **UML** and **ER.** A good idea is to look at **UML OCL** [34] as **CVSL** since working in this line would make contributions towards a *standard* approach. Note that we are not looking into [6] as an approach to ontology views. Such approach may lead to the extension of representation systems itself and this requires radical changes (formalism changes) on top of that language. We prefer to leave the ontology structure as it is and then require **CVSL** to derive classes which lead to the creation of ontology views.

We argue that working in this direction would fit well with a traversal-based paradigm as similar works proven in [23]. Nevertheless, we are not going to discuss this further since outside of this paper.

5.3 Requirement (R3) – The Views Require a "Well-Founded" System

To address **P3**, our **R3** requires us to employ a "*system*" that has a well-founded ontological representation. The closest works on this direction are efforts in formal ontology such as [35, 36]. In fact in [37] state that the Semantic Web relies heavily on formal ontology to structure data for comprehensive and transportable machine understanding. The necessity for a well-founded system is proven by a well-known works called the OntoClean [38], which is recognized as a prominent *"theoretical tool"* that useful to *"normalize"* the ontology. Furthermore, to contribute to the standard approach in ontology engineering, [24] has adopted some works from [38] to improve some structural aspects of a well-known formal ontology, the **BWW** [36].

We argue that research on ontology views should utilize formal ontology as an upper-level system that may *organize* and *structure* domain knowledge prior to any transformation to the *"practical"* traditional languages. We mean *"practical"* here is to refer to the conventional (traditional) systems either standard graphical ontology language like **UML ODM** or various *machine-readable* languages from **XML** to **RDFS** and leading up to standard semantic web language like **OWL.**

5.4 Requirement (R4) – The Views Require a "Publish/Subscribe" Mechanism

To approach **P4**, our **R4** requires we employ a *"mechanism"* that could manage users and owners for using interlocking ontologies structure. In fact, this mechanism will assist us for the maintenance of interlocking ontologies. At the individual-level, every owner is generally responsible for creation of some institutional facts (performative speech acts), and for interoperation other players (users or other owners) need to know the facts created (informative speech acts).

The informative speech acts can be either *"push"* or *"pull"*. A system of *"push"* and *"pull"* is active and modeling this kind of system really needs a credible modeling language. We argue that to fulfill this requirement, it is crucial that approach to ontology views require modeling methods at conceptualization-level.

5.5 Requirement (R5) – The Views Require a "Boundary" Mechanism

To approach **P5**, our **R5** requires we employ a *"mechanism"* that could facilitate users to *"cut off"* a view (an extract) that he or she wants. This reflects to the need for the user to have a *"specific"* or *"generic"* traversal over an ontology graph in order to have a *minimal* set of ontologies. This mechanism would give a kind of *"coverage"* to scope ontological dependency among ontology objects (e.g., classes) along the extraction process. There are several related works that we could have some ideas from. Some of them are published in [8, 11].

However, we argue that the boundary mechanism in such works do not treat ontology objects which are *mandatory* (e.g., specific or generic dependency) or *optional* (e.g., is not necessarily dependent) in a proper way. Our intention is to look at these two notions of ontological dependence as our guidance to have a minimal set of ontologies and with respect to the boundary mechanism.

5.6 Requirement (R6) – The Views and a "Layered" Approach

We need a *broader* thought to address all the proposed requirements. Therefore, we define our final requirement, **R6** which is *a layered approach* where similar works done in [16, 23]. The layered pattern is remarkable modeling approach primarily in **SE**. Fig. 3 illustrates our *conceptual framework* as solution for our ontology views. Unlike [23], our framework generally consists of two layers defined as a *Conceptualization-Specification Layer* (**C-S** Layer). Furthermore, we adapt definition in [16] to further decompose a *specification layer* into another three layers; *Application-Semantic-Syntactic Layer* (**A-S-S** Layer).

We further decompose our application-layer into Conceptual Views Layer (**CVL**) and Instance Views layer (**IVL**) (not shown in the Fig. 3 for the sake of clarity) which is slightly similar to [23]. Our **CVL** relates to the ontology views which are represented in formal ontology, while IVL concerns on a corresponding view (resulted from transformation process) represented using *"practical"* language like **OWL**.

In other words, we are going to introduce a more *comprehensive perspective* to ontology views. From a specification-layer, we have a similar idea with [16, 23] but in different way of point of views. However, we definitely differ from them and other approaches [5-19] since we introduce another layer so-called the *conceptualization-layer*. A detailed discussion on these layers including its advantages and benefits compared to other approaches is not the scope of this paper. This paper is only concerned on how our problems and requirements will be addressed in the context of our conceptual framework.

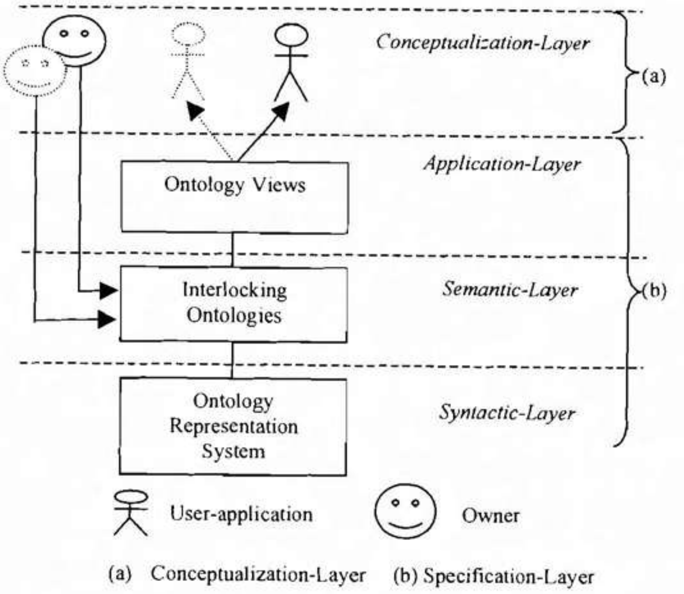

Fig. 3. A conceptual framework of our ontology views in **IWs**.

A layer where we develop a *"real world" (semantic)* model consists of terms, relationships and constraints of a given domain (e.g., the **IWs** involves constraints), we call this level the *semantic-layer*. A layer where we can define/use *metamodel*, in our case it is built-in *constructs* given in formal ontology, represents the *syntactic* specification of the ontology (semantic model). We term this level the *syntactic-layer*. Creating a view or an extract of the ontology with respect to the user's perception, we call this level the *application-layer*. Note that, our specification-layer (semantic and syntactic) supposes fully based on formal ontology (see **R3**) and could be transformed to the *"practical"* systems like **OWL** at application-layer (transformation between **CVL** and **IVL**).

Finally, the topmost level is the *conceptualization-layer*. This is a special level since the general idea is to get a *"business model"* of how users as well as owners *coordinate* their works for using interlocking ontologies structure. Briefly, with respect to our framework, the **P1** will be addressed by **R1** (conceptualization-layer), while the **P3** and **P5** will be addressed by **R3** and **R5** respectively (specification-layer). For the **P2** and **P4** which are respectively addressed by **R2** and **R4** will relate to either layers, or simply the **C-S** layer.

6. DISCUSSION

In [23] classifies research on views model into four categories, namely; (i) classical (relational) views, (ii) Object-oriented view models, (iii) semi-structured (namely **XML**) view models and (iv) view models for Semantic Web (**SW**). This paper relates to view models for **SW** where the closest works are presented from the research in ontology modularization techniques as surveyed in [8, 39]. In [8], such techniques are distinguished into at least three *paradigms*: (i) query-based, (ii) partition-based and (iii) traversal-based. We add a fourth paradigm into the survey literature as termed here (iv) extension-based such works in [6].

Briefly, the query-based inspired from database field where language alike **SQL** is used to define views, while the partition-based decompose the ontology graph into modules (views). In contrast, the traversal-based does not destroy the ontology and leaves the structure of the original ontology intact: it creates an extract, not composition. Finally, the extension-based proposes an extension to a particular representation system like **OWL** in order to define views. For example, [6] extends **OWL** classes and expressions to model ontology views as first-class objects.

As explained in section 4.5, our notion on ontology views is going to be based on traversal paradigm. We are going to differ from others in the following properties: **(i) Application and Problem Domain**; Most ontology modularization approaches are developed independently from problem and application domain as argued in [28]. Our ontology views problems is definitely clear. It is purposely for the context of **IWs** with a particular application domain; ontology server generating ontology views for supporting ontology-based interoperation of information systems.

This application domain falls into a *commit-time phase* in the context of ontology server usage classified in [2], and in fact most ontology servers developed for a design-phase [40]. In other words, we have a clear goal on ontology views so would lead to a proper evaluation; **(ii) A Layered Approach;** a layered approach is common in **SE** modeling to realize the notion of *separation of concern* [41]. It has many advantages such as separating implementation issues from conceptual aspect of modeling ontology views, such work is done in [23]. With different views to [16, 23] our conceptual framework aforementioned has introduced a new layer called a conceptualization-layer derived from the ontology definition [1] and elaborated in [20].

To our knowledge, we do not come across any ontology views mechanism addresses this sort of layer. In our context, this layer is important because interoperation may cause a *semantic heterogeneity* which needs to be resolved at this level [20]. From modeling point of view, we argue that some *meta-concepts* need to be captured from this level. By definition, our specification-layer's components is adapted from [16], however, we argue the idea to adapt **ANSI SPARC** [42] to the area of ontologies in [16] is not appropriate. For example, **ANSI SPARC** physical layer involves the layout of data on the disk, the indexes and so on. In [16], this corresponds to the physical schema represents the syntactic specification of the ontology (e.g., built-in constructs (metamodel) given in **OWL**).

Conversely, we argue in the **ANSI SPARC** system this probably also conceptual. To our knowledge, analogously to **MDA** [43] layers is much more appropriate rather than **ANSI SPARC.** For example, the syntactic-layer would correspond to the **MDA M2** (Metamodel-Layer) since this layer concerns on metamodel to represent ontology. Our semantic-layer would correspond to the **MDA M1** (Model-Layer) since this layer represents the terms and principles e.g., constraints of the domain (e.g., an Olympic ontology) which is represented by using e.g., metamodel from the syntactic-layer.

Our application-layer concerns to the aspect of *"real world"* including *behavior* and *structural* (e.g., **UML** activity diagram and **UML** classes where users extract views) as well as focusing on modeling views, the **CVL** (e.g., an extract of classes and associations) is **MDA M1** while the **IVL** is **MDA M0** (Instance-Layer). **(iii) Representation System;** most approaches to ontology views basing on conventional representation systems like **OWL** and some other approaches adopting visual representation systems like **UML** and **UML ODM** to benefit from many standard modeling tools for ontology. However, these conventional systems suffer from ontological semantics [24, 44]. For example, those systems do not contribute well in ontological dependence.

Therefore, we argue that conventional systems [25], [26], [27] or [22] are not appropriate for our semantic and syntactic layers. However, they are important for our application-layer (e.g., transformation between **CVL** and **IVL**); **(iv) Approach to Ontology Maintenance;** our thought of *"private"* and *"public"* may lead to the need for a new method to maintaining ontology via publish/subscribe mechanism.

This mechanism fits to our context. To our knowledge, we have not encountered this particular mechanism to ontology views; **(v) Ontological Dependence;** dependence is a varied notion. It is important notion in ontology representation [44]. The property Student, for example, is dependent, since to be a student there must be a teacher; for every instance of student there is at least one instance of teacher. Unlike others, our approach to ontology views will take this notion seriously as addressed significantly in the formal ontology like [24, 38].

7. CONCLUSION AND FURTHER WORKS

In the context of **IWs** and towards a *standard* approach to ontology and ontology server, this paper contributes into twofold: (i) several *problems* and *requirements* as a proposal for our ontology views; (ii) The *conceptual framework* as a conceptual solution to address these requirements with respect to our problems. Both are our *preliminary* findings derived at *high-level analysis*. As a result, we have a good *insight* or *key* ideas prior to the realization of our ontology views mechanism (e.g., at logical-stage and implementation-stage solutions).

For future works, our conceptual framework is essential as a *"general blueprint"* to *formally* specify our approach to ontology views. Furthermore, the specification for ontology views should be integrated in a concrete application domain; the ontology server generating ontology views. So, the server can represent the selected relevant parts of the ontology as *views* to the user's perspectives.

REFERENCES

1. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5** (1993) 199-220

2. Colomb, R.M.: Ontology and The Semantic Web IOS Press, Berlin (2007)

3. Ahmad, M.N., Colomb, R.M.: Commit-Time Requirements for an Ontology Server. 11th Pacific-Asia Conference on Information Systems (PACIS'07), Auckland, New Zealand (2007) 162-172

4. Ahmad, M.N., Colomb, R.M., Cole, J.: An Ontology Server: A Knowledge Tool for Systems Interoperability in Semantic Web Context. Malaysian Journal of Information Technology **18** (2006) 1-23

5. Volz, R., Oberle, D., Studer, R.: Views for lightweight web ontologies. Proceedings of the ACM Symposium on Applied Computing (SAC'03) (2003)

6. Uceda-Sosa, R., Chen, C.X.; Claypool, K.T.: CLOVE: A Framework to Design Ontology Views. Conceptual Modeling (ER'04) (2004)

7. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan (2004)

8. Seidenberg, J., Rector, A.: Web Ontology Segmentation: Analysis, Classification and Use. International World Wide Web Conference (WWW'06). ACM Edinburgh, Scotland (2006)

9. Seaborne, A., Prud'hommeaux, E.: SparQL Query Language for RDF. (2005)

10. Rector, A.: Modularization of domain ontologies implemented in descriptions logics and related formalism including OWL. International Conference on Knowledge Capture Island. ACM Press, Florida, USA (2003) 121-128

11. Noy, N.F., Musen, M.A.: Specifying Ontology Views by Traversal. International Semantic Web Conference (2004) 713-725

12. Magkanaraki, A., Tannen, V., Christophides, V., Plexousakis, D.: Viewing the Semantic Web through RVL Lenses. Journal of Web Semantics 1 (2004) 29

13. MacCartey, B., McIlraith, S., Amir, E., Uribe, T.: Practical Partition-Based Theorem Proving for Large Knowledge Bases. International Joint Conference on Artificial Intelligent (IJCAI'03) (2003)

14. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL Ontologies. Proceedings of the KCAP-2005 Workshop on Ontology Management, Banff, Canada (2005)

15. Grau, B.C., Parsia, B., Sirin, E.: Combining OWL ontologies using e-connections. Journal of Web Semantics 4 (2005) 1-42

16. Gahleitner, E., WoB, W.: POVOO - Process Oriented Views on Ontologies. Semantic Business Process Management (SBPM'06), Budva, Montenegro (2006)

17. Doran, P.: Ontology reuse via ontology modularisation. KnowledgeWeb PhD Symposium (KWEPS'2006), Budva, Montenegero (2006)

18. d'Aquin, M., Sabou, M., Motta, E.: Modularization: a Key for the Dynamic Selection of Relevant Knowledge Components. First International Workshop on Modular Ontologies (WoMO-2006), collocated with 5th International Semantic Web Conference (ISWC'06), Athens, GA, USA (2006)

19. Bhatt, M., Wouters, C., Flahive, A., Rahayu, W., Taniar, D.: Semantic Completeness in Sub-Ontology Extraction using Distributed Methods. In: Lagana, A., Gavrilova, M.L., Kumar, V. (eds.): Computational Science and Its Application, Vol. 3045. Springer-Verlag GmBH (2004) 508-517

20. Colomb, R.M., Ahmad, M.N.: Merging Ontologies Requires Interlocking Institutional Worlds. Journal of Applied Ontology **2** (2007) 1-12

21. Searle, J.R.: The Construction of Social Reality. The Free Press, New York (1995)

22. OMG-ODM: Ontology Definition Metamodel Final Adopted Specification. (2006)

23. Rajugan, R., Tharam, S., Dillon, T.S.: Modeling Views in the Layered View Model for XML Using UML. Journal of Web Information Systems **2** (2006) 95-117

24. Guizzardi, G.: The Role of Foundational Ontologies for Conceptual Modeling and Domain Ontology Representation. 7th International Baltic Conference on Databases and Information Systems, Vilnius, Lithuania (2006)

25. Chen, P.: The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems **1** (1976) 9-37

26. OMG-UML: Unified Modeling Language (2004)

27. W3C-OWL: OWL Web Ontology Language Overview (2007)

28. d'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Ontology Modularization for Knowledge Selection: Experiments and Evaluations. DEXA, Lecture Notes in Computer Science **4653** (2007) 874-883

29. Dietz, J.L.G.: The Deep Structure of Business Process. Communication of the ACM **49** (2006) 59-64

30. Weigand, H.: Two Decades of the Language-Action Perspective. Communication of the ACM **49** (2006) 45-46

31. Rittgen, P.: A Language-Mapping Approach to Action-Oriented Development of Information Systems. European Journal of Information Systems **15** (2006) 70-81

32. Dietz, J.L.G.: Deriving Use Cases from Business Process Models. ER 2003, LNCS 2813 (2003) 131-143

33. Colomb, R.M.: The Object Management Group Ontology Definition Meta-Model. In: Ruiz, F., Piattini, M. (eds.): Ontologies for Software Engineering and Software Technology. Springer-Verlag, Berlin (2006) 217-247

34. OMG-OCL: UML 2.0 OCL Final Adopted Specification (2003)

35. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. University of Twente, Telematica Instituut, Netherlands (2005)

36. Wand, Y., Weber, R.: On the Deep Structure of Information Systems. Information Systems Journal **3** (1995) 203-223

37. Maedche, A., Staab, S.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems and Their Applications **16** (2001) 72-79

38. Guarino, N., Welty, C.: An Overview of OntoClean. In: Staab, S., Studer, R. (ed.): Handbook on Ontologies. Springer-Verlag (2004) 151-159

39.Rector, A., Napoli, A., Stamou, G., Stoilos, G., Wolger, H., Pan, J., D' Aquin, M., Spaccapietra, S., Tzouvaras, V.: Report on modularisation of ontologies. Technical Report, Knowledge Web Deliverable D2.1.3.1 (2005)

40. Ahmad, M.N., Colomb, R.M.: Overview of Ontology Server Research. Journal of Webology **4** (2007) 1-7

41. Maciaszek, L.A.: Requirements Analysis And System Design. Addison Wesley, UK (2001)

42. Tsichritzis, D., Klug, A.: The ANSI/x3/sparc/dbms Framework Report of the Study Group on Database Management Systems Information Systems **3** (1978) 173-191

43. Mellor, S., Scott, K., Uhl, A., Weise, D.: MDA Distilled, Principles of Model Driven Architecture. Addison-Wesley Professional (2004)

44. Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. Communication of ACM **45** (2002) 61-65