MODELING CONVENTIONAL ARCHITECTURAL PROCESSES

Maintaining knowledge created by institutions situated in a historic context

ANAND BHATT
ABA-NET/Architexturez IMPRINTS
anand.bhatt@ab-a.net

Abstract. In this paper we present a knowledge representation infrastructure that is being tested in collaboration with several institutions. This infrastructure is designed to map the structure of institutions dealing in Architecture and related disciplines, their interrelationships, and knowledge constructed by institutionalized processes in a given historical context.

Keywords. Knowledge Acquisition, Modelling, Knowledge Storage, Inferencing, Portals

1. Context

Architectural processes at a are conventional processes. They are constructed by competent, statutory organizations in a networked communication across space and time. Statutory organizations operate at a formal level, where they have evolved norms, and *frames* or data-structures for representing stereotyped situations as encountered in practice (Minsky 1974). These frames represent the expertise of each organization and constantly adapt with the architectural process in general, and realities at large.

Such a communication necessarily occurs on a *historic* register as opposed to a discursive one: organizations encode architectural processes with the frames through exercises of power in their provenance, and in doing so, they encounter frames instituted by their peers. At an *epistemological,* level, they align their frames with processes that have emerged for reasons of consistency. At a *conceptual* level, they deploy the frames to execute tasks so as to realize institutional aims. Such processes have energized within Indian architectural processes in recent years, and a perceivable breakdown in the conventional architectural processes occurs as newer institutional formations and organizations of labor are rapidly inducted in practice,at several levels.

In this paper we describe an knowledge technologies platform designed to circumscribe such a situation. The platform provides a knowledge modeling environment, and an internet-based architecture to work with the modeled knowledge. Institutions can use the system to assert knowledge, analyze their capabilities and create scenarios to visualize their goals and perspectives.

2. The System

We need to circumscribe a system that resides between an epistemological level and a conceptual one. Guarino, (1995) describes such a level as an ontological level, where “a KR formalism is constrained in such a way that its intended models are made explicit”, as the goal is, "to restrict the number of possible interpretations, characterizing the meaning of the basic ontological categories used to describe the domain." Ontological levels are not transfer levels, and our knowledge representations are not repositories of knowledge extracted from an expert's mind. They are the result of a modeling activity whose object is the observed behavior of intelligent agents and organizations embedded in an external environment.

An ontological level is therefore at a level of meaning, more related to the classical notion of truth intended as correspondence to the real world, and less dependent on the particular way an intelligent agent pursues its goals. Clancey (1993) arguing that "the primary concern of knowledge engineering is modeling systems in the world, not replicating how people think", proposed the Modeling View, consisting of correspondence between a knowledgebase and two separate subsystems, representing the agent's behavior (problem-solving expertize) and its own environment (problem domain).

Actors engaged in conventional architectural processes in India already have *a-priori* understandings of their problem domains, and most of these understandings are recorded. Understandings may be encoded law. They are obtained in communicative action resulting in charters, conventions and instruments produced by organizations, and created by pedagogical activities aiming to transmit the function and field of the discipline. At a conceptual level, *a-posteriori* understandings are brought about through a asynchronous communication at a historic level as organizations engage with one-another through quasi-legal processes through exercise of power and test their definitions of the problem domain. This gives raise to a very organic situation from a modeling point of view as actors engaged who define the problem domains, who are ostensibly the consumers of modeled knowledge are also behavioral agents capable of building the knowledge, and the system attempting to circumscribe the problem domain must rely on the actors who are, in theory, its subjects. This results in a classic situation where the consumption of knowledge will have resulted in creation of knowledge, and the "system" could only mediate amongst the actors at a semantic level.

As a result, the system is designed in two parts, which are, (i) a knowledge-services consumer part, tasked with modeling the subsystems, transactions such as the creation of knowledgebases and knowledge extraction; and (ii) a knowledge-services provider part, tasked with maintaining the ontological level, truth-maintenance and storage of knowledge-objects that are created in the course of transactions.

3. Knowledge-Services Provider

The provider has three components, one, Knowledge Storage mechanisms to access knowledge, two, services to maintain modeled knowledge, and three, services to classify instances and compose conditional models from stored models and dispatch the compositions to the consumer.

3.1. ACCESSING THE KNOWLEDGE MODELS

The Knowledge Models conform to description-logic compliant species of Web Ontology Language, OWL-DL, selected for its ability to embody knowledge while retaining maximum expressiveness, computational completeness and decidability while reasoning.

3.1.1. Stored Knowledge Subsystems

Knowledge models are stored in a storage container for RDF (Sesame, Broekstra1, 2002) Sesame provides a Storage and Inference Layer, which is an internal API that abstracts from the storage format used and provides reasoning support. On top of the SAIL are functional modules, such as the semantic web endpoints and RDF export accessed through an Access APIs.

3.1.2. Stored knowledgebase

The knowledgebase incorporates a set of OWL ontologies, the Architexturez South Asia Corpus (http://architexturez.net), developed in engagement with Indian institutions. It is in a RDBMS accessed via the Sesame's access API..

3.1.3. Maintaining the Knowledge

Knowledge is maintained by stakeholders via various acts of knowledge-consumption outlined below. Each stakeholder is provided with tools to update OWL constructs. The Architexturez Platform provides for text processing, automated knowledge extraction and ontology learning.

3.1.3. Documentation and tests

UML models of the system, documentation, issue trackers and unit tests can be accessed via http://portal.architexturez.org.

3.2. MAINTAINING THE KNOWLEDGE MODELS

Knowledge maintenance are they are responsible for defining our subsystems in a way that resemble the exterior in a meaningful way, where it remains feasible to maintain truth within our models at the required levels. The knowledge model consists of thousands of frames describing the problem domain. These frames are provided by institutions or captured by observing conventional processes as they unfold in reality. Knowledge maintenance tasks emerge as institutions move towards unfamiliar situations where the stereotyped situations may supply fine details, or more analysis is needed. A series of correspondences between an ontological level arise at this moment as new facts are supplied and our reasoning facilities create new predicates.

3.2.1 Defining the Subsystems

We define the knowledge models for two distinct cases, first, as a representation of the problem domain, that is, "a model of entity and relationship in a specific domain or universe of discourse (UoD)" (Fresnel, 2003) and second, as modeled behaviors, or "specification of a conceptualization" (Gruber, 2008).

The system provides a UoD in the first case. It comprises of of several ontologies containing models for specific sub-domains of activity as constructed by the institutions, and a Role Metric to contextualize each part of the model to the other part. Several controlled vocabularies representing specific problem domains are included, and a user can configure his view of the problem domain to best reconcile with his situation by loading the models.

The system models conceptualizing tasks as encountered by institutions in the second case. They map conceptualizations, which are abstract process-patters based on cognitive engineering patterns of "particulars in the sense that the UoD is restricted to them", providing just "descriptive notions that assist in making already formed conceptualizations explicit" (Pisanelli, 2002).

3.2.2. Modeling the Subsystems

Figure-1 shows the primary entities in context, which are (i) Legal Agent, or an individual whose existence is granted simply by its communicability and capability of action. having Role Metrics represented by Information Realizations (ii) Concept Schemes, representing the required abilities, (iii) Concrete Roles, or an institutionalized capability representing an Abstract Role; (iv) Information Realizations, or processes representing transactions between agents, concepts and things found in reality, and (v) Concept Schemes, expressed through taxonomies, thesauri or ontologies indicative of a certain UoD, conceptualization or processes found in the physical world.

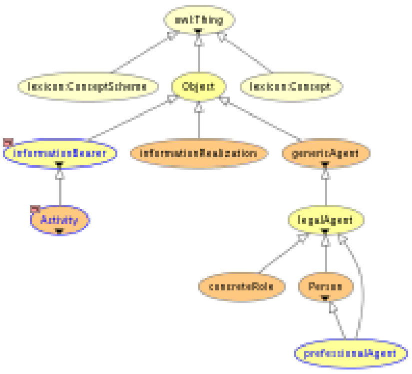

Figure 1: Asserted Structure of the Entities

3.3. REASONING THE KNOWLEDGE MODEL

The primary entities are supported by several secondary entities introduced to formalize the relationships and reason with the entities using several alternate pathways [Figure-2]. The reasoning capabilities of the knowledge-services provider are limited to tasks such as ontology evolution, truth maintenance and deterministic debugging of OWL models, as exemplified below.

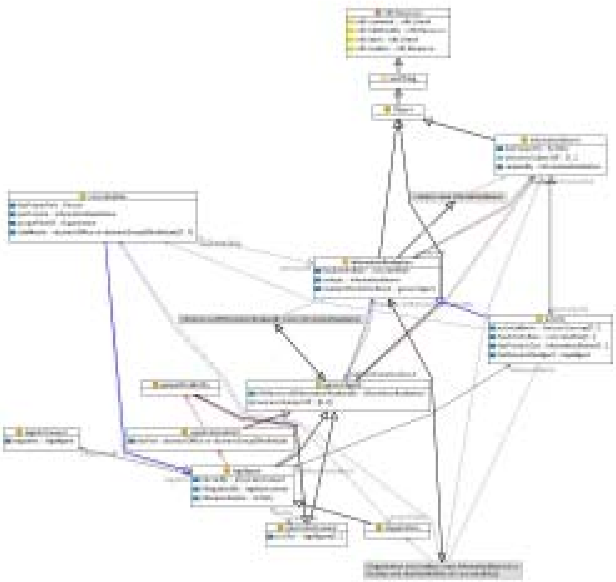

Figure 2: Secondary constructs in the Immediate Neighborhood of the Primary Types

3.3.1 example: professionalAgent

This example shows class and instance inferences resolving the status of a professionalAgent (Figure-1) in two distinct concept schemes. On one hand, the Council of Architecture in India, (CoA) asserts his hierarchical relation with other operators in the field. On the other UNESCO, has a category “conservator” that does not recognize this hierarchy. As the two organizations communicate through a network of international conventions the two assertions need to be resolved at an ontological level.

Table 1 defines the professionalAgent with the following conditions: (i) The Agent must not be the subject of any role metric, and the system must infer his role, (ii) The Agent must have an educational qualification that is regulated by the CoA, (iii) The Agent is responsible for at least one Information Realization recognized by the CoA or UNESCO, (iv) The Agent must have at least one concrete role or information realization, (vi) The Agent may be the subject of at least one information realization (viii) This Agent's function and field must disjoint with agents who are not architects. Resulting class inferences are shown in section 3.4 (below).

We can detail the model to a high granularity and create instance inferences by asserting, for example, that the professional is a conservator if (i) the agent's concreteRole, has a role metric in conservation architecture as defined UNESCO, or (ii) the agent has performed participated in information realizations characterized by UNESCO. The professional will be classed as a Conservation Architect if (iii) the agent has education qualifications that are regulated by the CoA.

TABLE 1: A Person is a ProfessionalAgent

```
Disjoint Classes
    artisan, enthusiast
Abstract Syntax
        SubClassOf(Person restriction(properPartOf allValuesFrom(concreteRole)))
        SubClassOf(Person restriction(isResponsibleFor minCardinality(1)))
        SubClassOf(Person legalAgent)
        Class(professionalAgent complete restriction(hasEducationQualification
        someValuesFrom(academicOrganisation))
            restriction(isRegulatedBy value(ArchitectsAct_1972)))
        Class(legalAgent partial genericAgent)
        isResponsibleFor MIN 1 owl:informationRealization
Equivalent Classes (Necessary and Sufficient Conditions)
        isReferenceOfInformationRealizedBy SOME (informationRealization
              restriction (inConceptScheme value (professionalArchitectScheme)))
Usage
        Class Description/Definition (Necessary Conditions)
        genericAgent, informationRealization, legalAgent, organisationalUnit,
        physicalnstrument, concreteRole, Organisation, Person, prefessionalAgent
Domain of
        isReferenceOfInformationRealizedBy
        isActedBy, isRegulatedBy, isResponsibleFor, graduatedFrom
```

3.3.2. example: concreteRole

This example shows some basic capabilities in truth maintenance. It shows how some concrete roles grant agents with capabilities which are alienable from the agent in a time-dependent manner [Table 2].

TABLE 2: A Concrete Role

```
Super Classes
        hasProperPart ONLY Person
Equivalent Classes (Necessary and Sufficient Conditions)
        legalAgent roleMetric EQ 1 owl:Thing
Abstract Syntax
        restriction(roleMetric cardinality(1))
            legalAgent
                    (restriction(properPartOf allValuesFrom(Organisation))
                    restriction(hasRoleMetric allValuesFrom(Architecture)))
            SubClassOf(concreteRole
                    restriction(hasProperPart allValuesFrom(Person)))
Usage
        Class Description/Definition (Necessary Conditions)
            concreteRole, informationRealization, Person Domain of participate, roleMetric
```

Concrete Roles have a transitive relation with an Abstract Office or an Abstract Group of Individuals, which is a generalization allowing for mappings between roles. Organizational Units are reusable patterns from which Organizations are created. [Table-3].

TABLE 3: An Abstract Office has a Transitive Relation with a Concrete Role

```
ObjectProperty(inRoleMetric inverseOf(roleMetric)
        domain(unionOf(abstractOffice abstractGroupOfIndividuals))
        range(concreteRole))
```

Concrete Roles are Legal Agents that populated by other Agents. They represent specific tasks executed by Agents in the course of their work, and are time-dependent; For example a person who is not a professional as determined by the CoA will be classified as a professional for the duration of his term embodies a role metric at this level.

3.3.3. example: informationRealization

This example elaborates the capability in truth maintenance. It shows how our system is capable of identifying a cognitive process, and placing other processes via class inheritance in context of the discovered process.

An information realization is a process representing the realization of an Information Bearing Object [Table 4]. It exemplifies our model's capability in embodying processes: information realizations evidence specific capabilities of various legal agents as they engage with realities. They bridge the three defining perspectives within our models, i.e., actors and legal agents; corporal things, and the frames created by organizations in the course of their work. At the most tangible level, information realisations are patterned after the Role Activity Artifact Tool Metric, providing "several types of metrics for various classes of things found in generic processes" (http://knowent.com).

TABLE 4: informationRealization and Adjacent Entities

```
Super Classes
    Object
    hasActivityRole ONLY concreteRole
        realizesInformationAbout ONLY genericAgent
        realizes SOME informationBearingObject
Equivalent Classes (Necessary and Sufficient Conditions)
    (legalAgent AND (realizes SOME informationBearingObject))
            OR (Activity AND (hasActivityRole ONLY concreteRole))
Abstract Syntax
    unionOf(intersectionOf(legalAgent
        restriction(realizes someValuesFrom(informationBearingObject)))
            intersectionOf (Activityrestriction(hasActivityRole
        allValuesFrom(concreteRole)))))
        SubClassOf(informationRealization
            (restriction(hasActivityRole allValuesFrom(concreteRole)
            restriction(realizesInformationAbout)) allValuesFrom(genericAgent)
        SubClassOf(informationRealization restriction(realizes)
            someValuesFrom(informationBearingObject)))
        SubClassOf(informationRealization
            restriction(realizes allValuesFrom(informationBearingObject)))
        SubClassOf(informationRealization Object)
Usage
    Class Description/Definition (Necessary Conditions)
    genericAgent, informationRealization
Domain of
    hasActivityRole, realizes, realizesInformationAbout
```

Although depicted as OWL Classes, Information realizations are processes that contain processes, and as such, they are critical to the knowledge maintenance tasks. Institutions and professionals will not be able to assert their status as legal agents if there were no information realization evidencing their capabilities in communicating.

3.4 COMPOSING A MODEL

A composite model [Figure-3] is created when the system reasons a more realistic situation from the assertions exemplified above, e.g, the assertion:

"A professionalAgent (IS-REGULATED-BY legalInstruments

(SUCH-AS acts-of-law)) TYPE-OF (legalAgent AND Person)"

is reified to a more realistic representation, [Figure-3], where the system has reasoned eliminated the hierarchical status of the professional, but the professional has role metrics that are unique to the profession:

"A professionalAgent (IS-REGULATED-BY legalInstruments

(SUCH-AS architect's-act-of-1972)) TYPE-OF Person

(GOVERNED-BY the legalInstrument)"

Similarly, the system has inferred that Activities are inheritance classes of Information Realization, i.e.:

"Activity TYPE-OF

(informationRealization and informationBearingObject)"

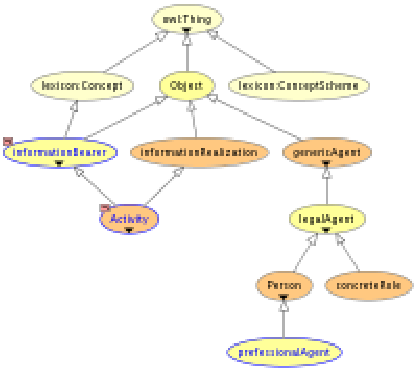

Figure 3: Inferred relationships between primary Entities

The system provides dispatchers, which communicate between the knowledge services provider and consumer. Dispatchers are a special type of reasoners that do not perform any reasoning by themselves, but routes goals to other reasoners. In our system they allow the knowledge consumers to reconcile reasoning processes with what they already recognize in their areas of competency, and in the process, constitute a system boundary between the knowledge services provider and consumers.

4. Knowledge-Services Consumer

The consumer tasked with three roles: first, construct and modify axioms within the knowledgebase, second, configure the models to best resemble the universe of discourse under processing, and third, query the knowledge. It is instantiated by institutions to retrieve models from the dispatcher.

These capabilities are not the same as model loading features commonly found on the semantic web. They rather assert a separation between ontology learning tools from ontology model in order for the services to be widely applicable and used. And then they drive partitions between different ontology models which coexist next to each other. Z-Object Publishing Environment (http://zope.org) knowledge-consumer allows for rapid prototyping of knowledge creation and retrieval workflows while adapting to the partitions required by knowledge consumption tasks. The Zope environment can interact with the Architexturez platform via an Unstructured Information Management Architecture (UIMA, http://incubator.apache.org//) and create Probabilistic Ontology Models (POM) by detecting corpus changes.

4.2 A KNOWLEDGE CREATION CASE

In this section we illustrate a typical use-case showing how the workflows make adaptations possible with our system: let us assume an organization needs to analyze a domain for complex tasks The organization has already collected the data it needs to process, and it has a corpus of texts defining its way of doing this; such as legal instruments and terms of engagement. Now the organization needs to instantiate a consumer to experiment with the data.

The organization retrieves a prototype workflow provided with default, “factory” installation of the consumer. The workflow are partitioned along three generic swimlanes, (i) a knowledge representation [KR] swimlane (ii) a knowledge acquisition [KA] swimlane, and (iii) an impact evaluation [IE] swimlane. Architexturez wll configure these to meet the requirements of the experiment underway. For example, the KR swimlane configured to represent the due diligence norms, the KA swimlane is populated with the ontologies representing the required data, and it invokes a UIMA architecture to mine the corpus of texts to generate a POM at runtime, and the IE swimlane is “told” about the XDI endpoints that will evaluate the impact of the experiment .

In the next step, the organization will run a set of simple services, where the KA component will invoke dispatcher and homogenize the data from trusted sources. For example, UNESCO will homogenize the data with the Archaeological Survey of India and International Council of Monuments and Sites. The user will then interact with the system and request specific modeling views, so as to investigate the result of the merge operation.

The organization’s specialists would evaluate the results of the merge and homogenize operation by executing test queries. Typical queries would concern the scope and competencies of several Legal Agents in India. The specialists will first evaluate if they could reconcile the results to their area of competency with cases to derive conclusions about the expertise they are investigating. The results of the KA process will be forwarded to the KR component, where the experts will be able to generate reports and browse data. They then run the IE component to study the impact of the hybrid model, and map the instances which are affected as a result. The IE component will return a graph showing nodes where the concerns overlap with other data sources. The resulting graph is returned to the KA component, where the specialists analyze the graphs, create new predicates and harmonize the set of concepts. The predicates are then stored as a new model [Figure-4].

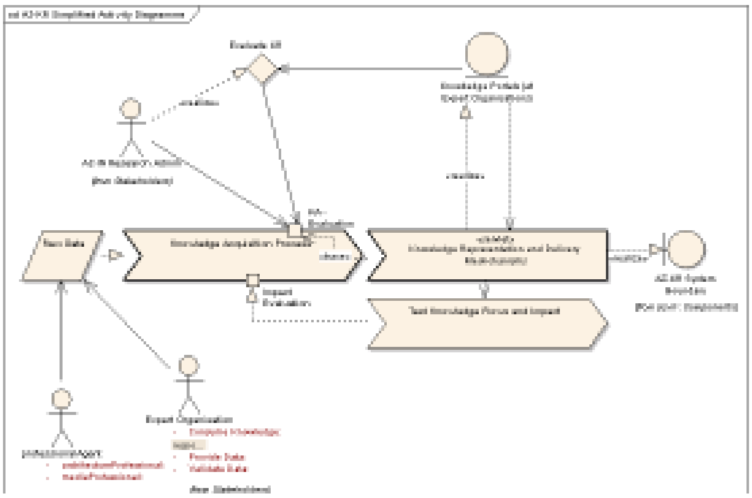

Figure 4: Prototype Activity Diagramme for Conditional Processing of Models

Finally, the organization will invoke the dispatcher in different ways, once satisfied, they will configure a client and publish the knowledgebase.

4.3 A KNOWLEDGE EXPLORATION CASE

The models are queried through the KR component via published knowledgebases where the experts can view compact descriptions of what remains unchanged throughout the history of the processes underlying the observed patterns. They can the knowledgebases using graph querying languages such as SPARQL. They can also invoke discovery engines through the client, and invoke several text-analysis and ontology learning tools. These tools extract the terminological knowledge embedded in the fetched documents and build POM at runtime. These operations facilitate experimentation with knowledge, and allow experts to evaluate the knowledge models. In historic cases, for example, they can evaluate semantics of change to study how the interpretation of a frame has transformed in time, that is, although the frame has remained invariant, the institutional dispositions surrounding it may have changed. This type of experimentation allows them to speculate on ramifications of a specific disposition or process [Figure-5].

A hard separation at software levels is maintained between models under evolution and published models. However, the user does not experience this separation in the knowledge consumption process. His experience of knowledge-consumption gives raise to an illusion of recursion between the creation and exploration of knowledge. This is an optical, even crucial requirement from the knowledge usability standpoint.

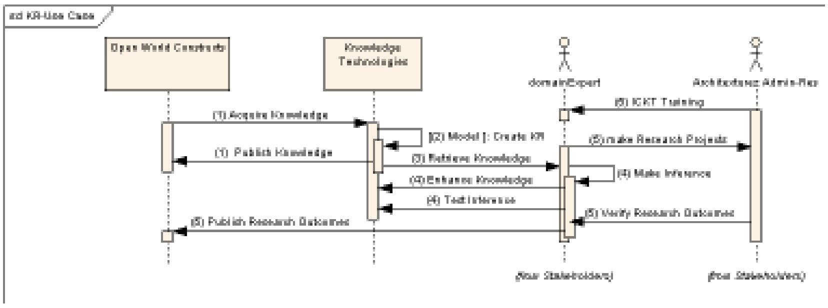

Figure 5: Prototype Sequence Diagramme for Conditional Processing of Models

5. Future Work

We have considered frames constituted by institutions so far, and defined conventional processes in architecture as processes that are conditionally obtained by knowledge creation processes that align (or communicate) the frames to one-another. We notice that organizations in India rely on a small pool of experts drawn from the field to formulate their policies. These experts also work independently of the institutions and in the process construct a networked communication amongst themselves at a molecular level. This communication is capable of influencing the inter-institutional, legislative and regulatory processes from ‘below’, and the affects of this type of a communication need to be studied further

We have so far resorted to some direct “short-path” characterizations in analyzing the agent’s behavior. In the next stages we will develop “long-path” characterizations. We will provide a discourse ontology with focus on argumentation moves and prototype analysis to assist in managing the complexity of a collaboratively built semantic graph of claims to this effect. This will provide a platform for principled disagreement in an open world situation. Following Boland (1995), this will necessitate “tools to track ideas and results in a field, and to express, analyze and contest their significance”.

We will focus on *affects* at the epistemological level where we will study affects as empowerments, and not a simple change or modification in the state of things. We will analyze how affects result in the creation of “blocks of space-time” at an ontological level, and are capable of inscribing architectural processes by themselves. Thrift (2004) has already demonstrated how the systematic engineering of such affects has become central to the political life of several cities, and therefore, to conventional architectural processes, and we need to find ways of expressing this phenomena at an ontological level while maintaining the tension between the various strands in the sciences and schools of philosophy that have guided us in our modeling activities.

6. Conclusion

In the course of development, we find a situation where the traditional objects of architectural thinking are made complex by additional tasks These tasks are about finding the locus of Legal Agents in architectural processes, where it becomes possible to ask questions about the capabilities, the function and field of each Legal Agent in a multitude of contexts. The tasks allow us to obtain a situation where the object architectural practice is no longer a matter of designing stuff in a world of concrete things, but rather finding the role of things in the system of abstract objects, where the aims of an architect's practice need to be communicated within an open world situation.

Our two-part system shows how it is possible to store a set of generic assertions about a given situation (the profession in India), and then consume them variously. We have created a system that allows institutions to experiment and visualize the conventional processes in architecture. They can study various configurations of agents, concepts and things in reality, or study the ramifications of bringing together a set of things in a specific way.

Acknowledgements

This project was executed in collaboration with Architexturez Network and UNESCO field office in New Delhi. It relies on previous work at Council of Architecture, New Delhi. The author is grateful to UNESCO Programme Specialists in Culture; Profs. Nalini Thakur, Akhtar Chauhan, Rohit Jigyasu, SANYOCOP and Mr Hemang Kulkarni for their assistance.

References

Boland, R. J. J., & Tenkasi, R. V. (1995). Perspective making and perspective taking *in communities of knowing. Organization Science*, 6(4), 350-372.

Broekstra, J. and Kampman, A., 2003: Inferencing and Truth Maintenance in RDF Schema. In Volz, R., Decker, S., and Cruz, I. (Eds.) *Proceedings of the First International Workshop on Practical and Scalable Semantic Systems*.

Broekstra1, J., Kampman1 A., and Harmelen, F.: 2002: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema *in* I Harrocks and J. Hendler (eds) *Proceedings of the International* Semantic Web *Conference 2002*, Springer-Verlag, Berlin, pp 54-68.

Clancey, W. J. 1993. The Knowledge Level Reinterpreted: Modelling Socio-Technical Systems. *International Journal of Intelligent Systems*, 8: 33-49.

Fensel, D.: 2003, Ontologies: *A Silver Bullet for Knowledge Management and Electronic Commerc*e, Springer-Verlag, New York.

Gruber, T. R.: 2008, Ontologies, *in* Ling Liu and M. Tamer zsu (Eds.), *The Encyclopedia of Database Systems,* Springer-Verlag, 2008.

Guarino, N. and Poli, R. (eds.) 1995. Formal Ontology in Conceptual Analysis and Knowledge Representation. *Special issue of the International Journal of Human and Computer Studies*, vol. 43 n. 5/6, Academic Press.

Minsky, M; 1974, A Framework for Representing Knowledge Reprinted in *The Psychology of Computer Vision*, P. Winston (Ed.), McGraw-Hill, 1975.

Pisanelli, D, Gangemi, A, and Steve G., 2002: *An ontology of descriptions and situations for Lyee's hypothetical world* , CNR – ISTC, Viale Marx 15, 00137, Rome, Italy

Thrift, N.: 2004: Intensities of Feeling: Towards a Spatial Politics of Affect *in Geografiska Annaler, Series B: Human Geography* 86 (1), Blackewll, 57–78.

UNESCO Publishing, 1995: *UNESCO Thesaurus: A Structured List of Descriptors for Indexing and Retrieving Literature in the Fields of Education, Science, Social and Human Science, Culture, Communication and Information.* UNESCO Publishing, Paris.