

MODELING CONVENTIONAL ARCHITECTURAL PROCESSES

Locating Expertise in the field of Heritage and Conservation in South Asia

FIRST A. AUTHOR

The Institute and its Address of the Author 1

Email address:

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

Indian architects today are operating on an unprecedented scale and complexity. We see hybrid building types and hybridized modes of practice. New divisions of labor and institutional formations are emerging where organizations have to work in fluid circumstances with regulatory and enabling institutions which are often in conflict with one-another. It is a situation where tasks emerge organically, and may not have a direct precedent, and where the data supplied by practitioners is not always reliable.

It is seen that standard methods used by institutions to locate experts in such as situation do not suffice given the complexity of the task that needs to be performed for various reasons. As a result, institutions that need to invest in Architecture find it difficult to execute several tasks, such as the choice of architects who may be capable of executing a given task, or estimating the capability of professionals to realize a certain type of institutional aim or perspective.

The Expertise Locator attempts to provide institutions with a tool which may be used in such circumstance. Institutions can use the system to measure competencies available in the profession, analyses the abilities of specific actors in the field, and create scenarios to validate their goals or perspectives.

2. The System

The system has two parts, (i) a knowledge-services provider part, tasked with the modeling and storage of knowledge-objects that are created in the course of transactions; and (ii) a knowledge-services consumer part, tasked with the creation of knowledge-models, and knowledge extraction. The consumer is created in an agile manner and it can be instantiated by institutions keeping their specific knowledge-extraction requirements in mind.

3. Knowledge-Services Provider

The knowledge-services provider has three components, one, Knowledge Storage – a mechanism allowing for the publishing and retrieval of knowledge by the knowledge consumers, the two, Knowledge-Models constructed to express expertise in Indian architectural processes, and three, Reasoning and Inferencing Services – to compose conditional models from stored models (see Section 4.2, below), as needed; and query the generated knowledgebase.

3.1. STORING THE KNOWLEDGE-MODELS

The knowledge-models are published on a subversion server as Web Ontology Language (OWL) files. OWL is a standard developed by the Web Ontology Working Group at W3.org. OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content. Most knowledge-models are written in the description-logic conformant species of OWL, called OWL-DL, and stored as RDF data for maximum interoperability with existing web clients.

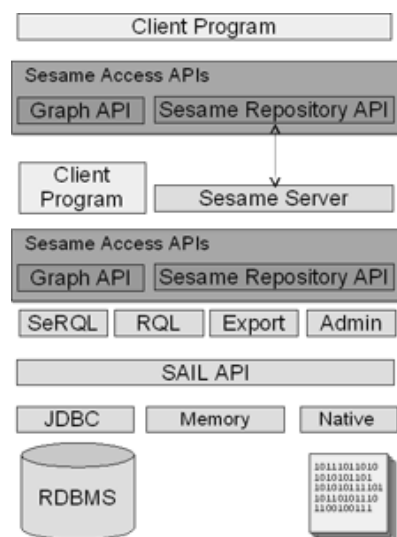


Figure 1: Sesame Server Architecture

Knowledge-models are stored in a set of Sesame (www.openrdf.org) [Figure-1] repositories, which are storage containers for RDF which may be accessed by humans using web browsers or API calls by client applications such as knowledge-service consumers.

Sesame provides a Storage and Inference Layer, or SAIL API, which is an internal Sesame API that abstracts from the storage format used (i.e. whether the data is stored in an RDBMS, in memory, or in files, for example), and provides reasoning support. SAIL implementations can also be stacked on top of each other, to provide functionality such as caching or concurrent access handling. Each Sesame repository has its own SAIL object to represent it.

On top of the SAIL, we find Sesame's functional modules, such as the SPARQL and RDQL query engines, and RDF export. Access to these functional modules is available through Sesame's Access APIs. [ref: Sesame Documentation]

3.2. THE KNOWLEDGE-MODEL

The knowledge-models rely on the South Asia Corpus Ontology, which was developed as a part of Architexturez-IN involvement with the regulation of education and research activity at Council of Architecture. The Corpus is further developed in collaboration with technical experts from UNESCO, and other organizations in India..

The first version of the models is in production use, and can be retrieved from Architexturez at www.architexturez.in. A set of OWL-Documentation is also provided on that URL. Development versions, API and User Documentation, research papers, tests and issue tracker and expert evaluations of the models can be accessed on <http://portal.architexturez.org>.

3.2.1 *Defining the Knowledge-Models*

We define knowledge-models alternately as (i) “ontology is a model of entity and relationship in a specific domain or universe of discourse (UoD)” and (ii) “specification of a conceptualization”. In the first instance of our definition, we provide Indian Institutions with a UoD consisting of several thesauri which in turn consist of models for specific sub-domains of Architectural activity as constructed by the institutions and a Role Metric which contextualizes each part of the model to the other part. In the second instance, we provide models of conceptualizing tasks as encountered by institutions. Conceptualizations are mapped in the knowledge-model Information-Realization tasks.

The current version of the knowledge-models expand patterns from Role Activity Artifact Tool Metric patterns, (<http://knoent.com/>), which is a set of general process pattern, utilizing “several types of metrics for various classes of things found in generic processes”. The models also use some patterns from A Descriptive Ontology for Linguistic and Cognitive Engineering (<http://www.loa-cnr.it/>), an ontology of “particulars in the sense that the UoD is restricted to them”, providing just “descriptive notions that assist in making already formed conceptualizations explicit”. The models also refer Actors and the NASA Expertise Locater.

3.2.2 *Designing the Knowledge-Model*

The knowledge-models are provide for multiple types of knowledge extraction tasks in locating expertise, based on the specific UoD under consideration (see Section 3.3, 4.2, below), and they are tasked with matching the required abilities with experts in the field.

At present the knowledge-model relies on three primary entities, (i) Legal Agent, representing a person or an organization in the field, (ii) Concept Schemes, representing the required abilities, (iii) Concrete Roles and Information Realizations, representing the mappings amongst the Legal Agent and Concept Schemes.

Figure-2 shows the primary entities in context, this is a generic model designed to support a single assertion: The expertise of a Legal Agent can be inferred from the Concrete Role played by the Legal Agent and the Information Realization processes the agent is a part of, where,

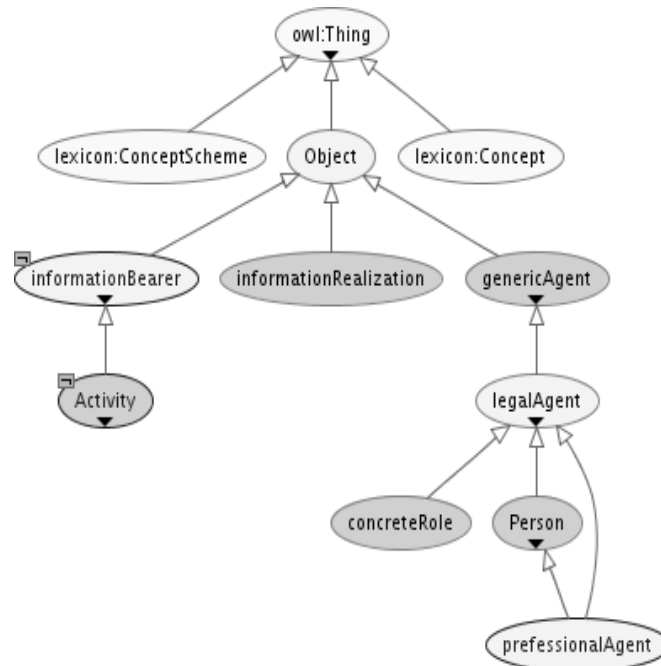


Figure 2: Asserted Structure of the Entities

1. A Legal Agent is defined as an individual whose existence is granted simply by its communicability and capability of action.
2. Concrete Roles are Legal Agents representing specific tasks executed by other Legal Agents in the course of their work,
3. Legal Agents and Concrete Roles have Role Metrics represented by Information Realization Processes
4. Information Realization Processes result in Information Bearing Objects, which are entities classed in Concept Schemes
5. Concept Schemes, are sets of concepts, optionally including statements about semantic relationships between those concepts and expressed via taxonomies indicative of a UoD or conceptualization.

The primary entities are supported by several secondary entities. These are introduced to formalize the relationships between the primary entities, and to define them in the given context. Figure-3 shows such a relationship.

The knowledge-services provider is nominally capable of inferencing (see Section 3.3, below). Inferencing is supported by modeling the attributes, restrictions and axioms between primary entities and secondary entities.

Super Classes	Person, legalAgent
	SubClassOf(genericAgent
	restriction(lexicon:isSubjectOf cardinality(0)))
	graduatedFrom SOME academicOrganisation
	isResponsibleFor MIN 1 owl:informationRealization
Equivalent Classes (Necessary and Sufficient Conditions)	isReferenceOfInformationRealizedBy SOME informationRealization
	properPartOf SOME concreteRole
Disjoint Classes	artisan, enthusiast
Abstract Syntax	Class(legalAgent partial genericAgent)
	DisjointClasses(legalAgent geopoliticalEntity organisationalUnit)
	SubClassOf(Person legalAgent)
	SubClassOf(Person restriction(isResponsibleFor minCardinality(1)))
	Class(professionalAgent partial restriction(graduatedFrom
	someValuesFrom(academicOrganisation)) Person legalAgent)
Usage	Class Description/Definition (Necessary Conditions)
	genericAgent, informationRealization, legalAgent,
	organisationalUnit,
	physicalInstrument, concreteRole, Organisation, Person,
	professionalAgent
Domain of	isReferenceOfInformationRealizedBy
	isActedBy, isRegulatedBy, isResponsibleFor, graduatedFrom

TABLE 1: The professionalAgent and Adjacent Entities

3.2.2.2. *entity: concreteRole*

Concrete Roles are offices occupied by Individuals or groups of individuals. They have more specific instances as they are parts of Organizational Things and have a transitive relation with an Abstract Office, which is a generalization allowing for mappings between roles. [Table 2].

We have decided that this is the smallest level at which causal links must be made between Tangible Things and Concepts so as to couple the Legal Agent hierarchy in an abstract pattern of things.

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

TABLE 2: concreteRole and adjacent entities

3.2.2.3. *entity: informationRealization*

An Information Realization is a concrete realization of an Information Bearing Object, e.g., a print document containing the text of a law. [Table 3].

Super Classes
Object
hasActivityRole ONLY concreteRole
realizesInformationAbout ONLY genericAgent
realizes SOME informationBearingObject
realizes ONLY 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)))
SubClassOf(informationRealization
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

TABLE 3: informationRealization and Adjacent Entities

3.3. REASONING THE KNOWLEDGE-MODEL

The knowledge-services provider limits itself to automatic inferencing that completes the knowledge-model. The provider executes some rather straightforward data-processing strategies to reify relationships amongst asserted models. This means that given a set of RDF and/or RDF Schema, the knowledge-services provider can find the implicit information in the data. Any advanced reasoning or inferencing task will be executed by the knowledge-services consumer, explained in Section 4.2, where knowledge-extraction is done based on contexts, concept-schemes and ontologies as entailed by the specifically instantiated client, for a specific use-case.

Taking asserted relations between the primary entities as an example, Figure-4 illustrates the inferencing abilities of the provider. Where the condition asserted in the model [figure-2], i.e.,

“A professionalAgent (IS-REGULATED-BY legalInstruments
(SUCH-AS acts-of-law)) TYPE-OF (legalAgent AND Person)”

is reified to a more realistic representation:

“A professionalAgent (IS-REGULATED-BY legalInstruments
(SUCH-AS architect's-act-of-1972)) TYPE-OF Person
(WITH contractual-responsibilities
GOVERNED-BY architect's-act-of-1972)”

Similarly, the system has inferred that Information Realization has a sub-type, Activity, i.e.,

“Activity TYPE-OF (informationRealization
and informationBearingObject)”

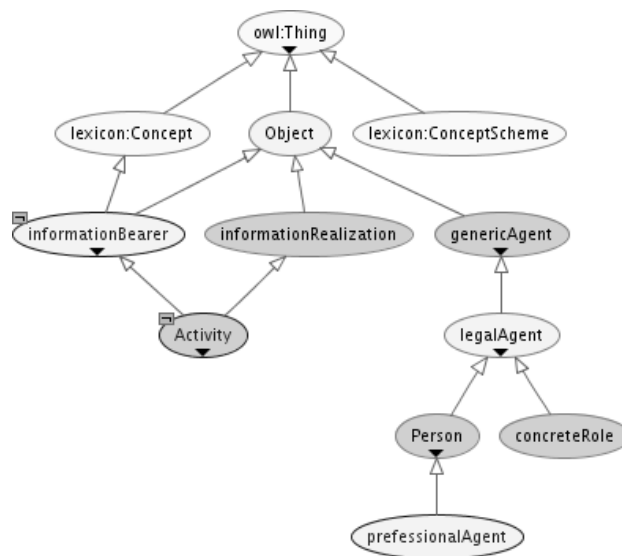


Figure 4: Inferred relationships between primary Entities

4. Knowledge-Services Consumer

The knowledge is tasked with three roles. It allows users to; first, construct the knowledge-models as needed, second, configure the knowledge-models to best resemble the universe of discourse under processing, and third, query the knowledge-services provider and extract knowledge. The knowledge-services consumer is provisioned for specific use-cases.

4.1. BUILDING THE KNOWLEDGE-SERVICES CONSUMER

Interfaces provided by the Sesame Repository API are combined for browsing and annotating the knowledgebase in the simplest instance of a knowledge-service provider. A typical knowledge browser will allow users to visually reconcile what they see with what they already recognize in their areas of competency. Several types of knowledge browsers are configured using portable descriptions of resources that use vocabularies for displaying RDF.

More complex instantiations allow client applications to configure the knowledgebase by loading specific models (see Section 4.2, below), or actively participate in the knowledge-generation tasks (see Section 4.3, below). A Z-Object Publishing Environment (Zope, <http://www.zope.org>) is maintained for more complex uses. Zope provides for an object database, and a pluggable architecture allowing for rapid prototyping of knowledge-creation workflows, knowledge management workflows, custom content-types and business logics of various types.

The knowledge-services consumer also provides some knowledge extraction and analysis tools, such as a similarity matrix extractors and a General Architecture in Text Engineering (GATE, <http://gate.ac.uk>), and statistical analysis packages such as Rapidminer (<http://rapid-i.com/>).

4.2. CONDITIONAL PROCESSING AND PLUGGABLE CONCEPT-SCHEMES

The knowledge-services provider is capable of inferring as to which representation bearers belong to a concept. It also provisions several concept-schemes specific task that needs to be evaluated.

The knowledge-consumer expands this capability and allows users to load various concept-schemes and the evaluate expertise of Legal Agents in various conditions. The knowledge-consumer also allows for rapid prototyping of knowledge creation workflows, views and methods to query the knowledge-provider [Figures-5 and 6]. A prototype workflow is provisioned in the Zope environment, it can be modified as required.

The organization already has collected the data it needs to process, and has Legal Instruments suggesting its models of practice; such as legal instruments, agreements and terms of engagement, and contracts. Now the organization needs to build a knowledge- consumer that can process our knowledge-models within the framework constituted by the organization.

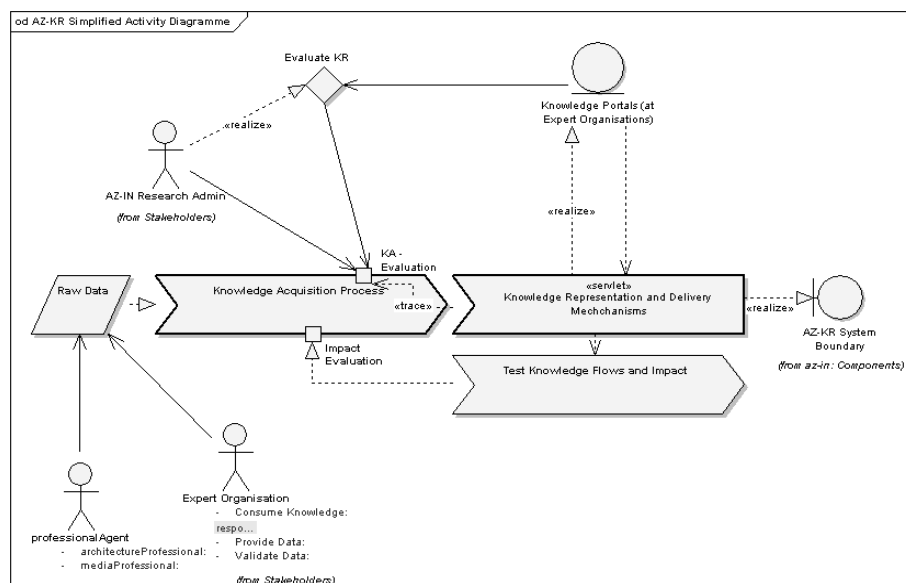


Figure 5: Prototype Activity Diagramme for Conditional Processing of Models

4.2.1 A Knowledge Creation Case

In this section we illustrate a typical use-case showing the adaptations possible with our system, where an intergovernmental organization needs to find domain experts for complex tasks.

The organization first retrieves three prototype process components provided with the knowledge-consumer. These components embody a generic knowledge acquisition [KA] process, a generic knowledge representation [KR] process and an impact evaluation [IE] process. The components are then modified to suit the business processes specific for an organization. For example, the KA process is modified to suit the peer review norms at the organization. Similarly, the IE process is customized to log the necessary information and generate reports customarily used by the organization.

In the next step, the organization would port all its data into the KA component, and identify trusted data providers. A set of simple services is configured to query the various data providers, and the KA component will invoke the knowledge-services provider. The knowledge-services provider will then infer relations and homogenize the data. This is a merge operation typical of the semantic web, where, for example, the UNESCO Office in New Delhi will homogenize the data with the Archaeological Survey of India, and International Council of Monuments and Sites. UNESCO would then request special views, so as to investigate the result of the merge operation. The KR component would create a simple knowledge browser in response.

UNESCO specialists would evaluate the results of the merge and homogenize operation by executing test queries. Typical test 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. Built heritage specialists, for example, may notice that in India Conservation Architects are regulated by a statute, and are professionals; whereas UNESCO classifies conservationists as vocationalists, or technical specialists. [Figure-5]

They will then identify the statements that cause the observed contradiction. That is, a Conservation Architect in India is an Architect, who is regulated by the Architect's Act (1972) in India, and going further, a previous KA process run by the Council of Architecture has already created a Concept Scheme about this Act.

```
conservationArchitect TYPE-OF professionalAgent:Architect
professionalAgent:Architect
    IS-REGULATED-BY legalInstrument:ArchitectsAct1972
legalInstrument:ArchitectsAct1972
    IN-CONCEPT-SCHEME
conceptScheme:ArchitectsAct1972
conceptScheme:ArchitectsAct1972
    HAS-TOP-CONCEPT Professional
```

UNESCO specialists will then reconfigure the KR component and load the UNESCO Thesaurus. as a complement of the Concept Scheme to better represent the formal definition of the Legal Agent in the UNESCO universe. 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 two Concept Schemes overlap, and

areas of discrepancy. Results are returned to the KA component, where the UNESCO specialists map instances between the two Concept Schemes, and harmonize the set of concepts. They may create new rules, restrictions, or state axioms to model a conservation architect in a way that satisfies both the Concept Schemes. The new mappings are then stored in the knowledgebase. [Figure-6]

In the final step, UNESCO may run more IE configurations by conditionally loading various Concept Schemes, and obtain various speculative scenarios to check how various institutions may collaborate by including their design capabilities or Concrete Roles, and the way institutions may interpret the same object.

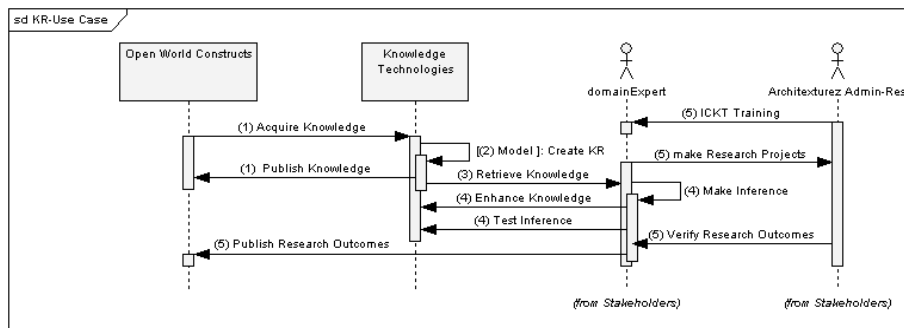


Figure 6: Prototype Sequence Diagramme for Conditional Processing of Models

4.2.2 A Knowledge Extraction Case

Town planning experts use existing documentation, and model processes showing the evolution of planning law Bye-Law and committee regimes circa 1860 in Delhi to result in a formal, modern institution. The models are queried through the KR component, and the experts can view compact descriptions of what remains unchanged in law throughout the history of the process (the frames) underlying the observed evolutionary patterns. They now want to look at specific instances of the frame. However, the knowledge-services provider has no such instance.

The experts can invoke a discovery engine, and query the legal database (JUDIS, <http://judis.nic.in>) to find case-laws pertaining the frame they have discovered. The discovery engine retrieves full text of court judgment and legal documents, which are then sent to the KA component.

The KA component has several text-analysis and ontology learning tools, which extract the terminological knowledge embedded in the fetched documents, and place it in context of the homogenized Concept Schemes.

The planning experts then evaluate the instances, and enter into a scholarly discussion about the instances. They evaluate semantics of change to study how the interpretation of the frame has transformed in time, that is, although the frame has remained invariant in time, the institutional dispositions and Concept Schemes surrounding it have changed in time.

Experts invoke the IE component, and graph the frame and its contexts. They run experiments on the graph, and analyze it. Their analysis allows them to visualize various planing processes, speculate on ramifications of a specific action, and otherwise act.

5. Conclusions

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 system of concepts. They can then organize Concept Schemes and Legal Agents in optimal configurations, or study the ramifications of bringing together a set of things in a specific way.

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 a profession, 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 places for things in the system of abstract objects, where the aims of an architect's practice need to be articulated within an open world situation.

Acknowledgements

UNESCO Field Office in New Delhi
Department of Conservation Architecture, SPA, New Delhi
Indian National Trust for Art and Cultural Heritage, New Delhi.
Council of Architecture, New Delhi.

References