Semantically Driven Architecture for Governance

Deepti[1], Rama Hariharan[2]* and Deepak Chand Misra[2]

ABSTRACT

A number of efforts in the area of e-Governance are being undertaken by the Indian Government to strengthen governance in the country. One area of governance to which not much attention has been paid is the area of modeling the government domain and providing tools and technologies for policy makers to directly update the government knowledge. The paper explores the approach of semantic technologies to address this aspect of governance based on the research efforts already taken up by different countries in this area. It briefly describes a research project that is under way to adopt this approach for the Indian government.

***Keywords**:* Government ontology, Government Web Services, Semantic Government Architecture

1. Introduction

The rapid development in the area of Information and Communication technologies (ICTs) and the proportionate reduction in the cost of hardware technologies in the past few decades has led to the widespread usage of these technologies. The governments across the globe have been making attempts to exploit the opportunities afforded by these technologies to improve the delivery of services to citizens while at the same time improving the efficiency, transparency and accountability of the Government itself. ICTs, if used properly, can act as a potent tool in facilitating good governance in the country. It can reach out to the most remote and marginalized sections of the society, promote transparency and accountability, enable swift delivery of information and services and increase productivity and efficiency. ICT can also introduce innovative ways in which citizens can interact with the government and participate in the process of governance.

The rise of E-Governance is a natural consequence of these developments. E-Governance is the application of ICT to exchange information and services with citizens, businesses and other arms of the government, in order to improve internal efficiency, effectiveness, transparency, accountability and better accessibility of services to citizens and businesses and to facilitate active participation of citizens in steering the governance process in terms of policies, programs and development planning. Information Technology (IT) interventions in the Indian government began during mid-1970s with the establishment of National Informatics Centre (NIC). At that time, the focus was primarily on computerization of isolated activities of the central government departments. Networked solutions were practically non-existent as communication technologies were just beginning to evolve in the country. The scene changed in mid-1980s when computers were deployed in all the districts of the country and networked through NICNET. Thus began the era of E-Governance in the country. Today, there are many aspects of governance, which have been

[1] B.Tech Student, IP University, New Delhi, India
[2] National Informatics Centre, A-Block, CGO Complex, Lodi Road, New Delhi - 110 003, India
* *Corresponding Author:* (E-mail : rama.h@nic.in, Telephone: +91-11-24360563)

automated and are helping to improve the life of citizens.

One problem with which everyone associated with e-Governance applications is aware of is the problem of interoperability. The development of software applications and hence the modeling of data have developed in a very decentralized manner. This has led to many models of data embedded in different software applications leading to inability of software applications to interoperate with one another and also leading to lack of re-use of domain knowledge. This could also be put in another manner: since there never was a central repository where government knowledge was modeled and stored, each software application was forced to create its own model and also embed the model and the associated rules in the software application. The problem can be best explained through an example.

2. The Domain of discourse – Developmental Governance

Here, we consider the domain of development governance. The development governance of India constitutes the most important part of governance in the country, both economically as well as socially. More than 80% of government exchequer is expended on development governance, thus underscoring its importance.

The entire cycle of development governance can be explained as follows:

- The Government defines a set of policies for each development sector.
- In order to implement these policies, the Government launches various programmes/schemes. The schemes are the sources of funding for various developmental works to be undertaken at the implementation level.
- Each planning unit at the grass root level prepares a draft plan based on the expected resource envelope under different scheme. The development works to be undertaken as part of the plan should ideally come through a participative process wherein citizens play a crucial role in defining their needs and aspirations.
- The plans from various planning units are consolidated at state and central government level. Having know the amount of financial resources that would be required, the Central government prepares the budget based on the available resources. The budget is broken down scheme-wise and the actual allocation made known to each planning unit that is authorized to receive the funds.
- Once the actual budget estimate is known, each planning unit can finalize its plan and prepare an action plan. All the developmental works are carried out as per the approved action plan.
- It is expected that the execution of the action plan would take the citizens to a better standard of living.
- The execution of the action plan is monitored and based on the impact on development, new developmental activities under the existing schemes may be taken up, or new programmes may be launched or existing schemes/programmes may be modified or even abandoned.

The above was a brief overview of the cycle of developmental governance adopted by the Indian government. The Schemes may be sponsored by the Central Government which would invariably be implemented across many states in the country or by the State Governments, in which case their implementation would be restricted to their individual states. The Central Government itself sponsors various types of schemes such as

- **Centrally Sponsored Schemes** where funding may be shared among the different stakeholders and execution is left to the state government
- **Central Sector Schemes** which are fully funded and executed by the Central Government and
- **Additional Central Assistance (ACA)** which are given as additional grants by the Central Government

Among these, the Centrally Sponsored Schemes are the most popular instruments of the Central Government. A number of scheme monitoring systems have been developed to facilitate the monitoring of the physical and financial progress of these schemes. Monitoring the proper implementation of these schemes is an important activity as a huge chunk of the government exchequer is spent on these programmes. Some examples of very good monitoring systems include those developed to monitor NREGS, ARWSP, Total Sanitation Campaign etc. Many more systems are under development for other schemes. Each of these systems has codified the rules and regulations defined in the scheme guidelines to ensure that the implementation does not go against the guidelines. In practice, the software designer goes through the guidelines, interprets the rules and regulations and builds them as part of the software application. The guidelines, being in natural language, are subject to mis-interpretation. The bureaucrat, who has prepared the guidelines and actually knows what he/she intended to mean in the guidelines, does not participate directly in capturing these details in the software. Instead, he participates through discussions with the software development team.

3. The Problem

Now, where is the problem? The problem is that each software application that needs to capture the business rules associated with a scheme encodes it in itself, either partly or wholly. Let us take three examples of software systems which may use the business rules specified in scheme guidelines:

- One could be a monitoring system to facilitate the government department, which has launched the scheme, to monitor the progress of implementation of the scheme.
- Another software application could be developed to help a citizen identify the schemes/ programmes for which she could be a potential beneficiary based on her socio-economic profile. Every scheme clearly specifies the profile of its intended beneficiaries.
- A third system could be a planning system which attempts to look for schemes under which funds would be available for taking up a developmental activity.

All the software applications obviously serve different purposes but are depended on the same source for their business logic viz., the scheme guidelines. Scheme guidelines are written in natural language and are, therefore, subject to a high level of misinterpretation. While on the one hand, misinterpretation could lead to invalid implementation of the schemes by government functionaries, on the other, it can lead to incorrect and varying specifications in different software systems. What, therefore, happens is that the same government knowledge is modeled in all the three different software applications, most probably differently. If the scheme is a central government scheme, then there is a possibility that several avatars of these software applications exist in each state, apparently to fulfill the state-specific needs. All of these lead to several problems:

- Government policies do change from time to time. For e.g., the target beneficiaries of a social welfare scheme could change or there could be changes to approval process of a plan. In the current scenario, any of these changes would impact each of the software applications that is dependent on the policy. Making changes to each software translates into cost as it involves deployment of additional resources in terms of man and man-months. If the changes are not reflected in all related software applications, it could also lead to inconsistency in the data.
- Since the data is most likely modeled differently in each of the applications, it would not be possible for these applications to interoperate.
- However, the larger problem is that the bureaucrat, who is the policy maker, is not directly involved in modeling the domain.

The current architecture of E-Governance is illustrated in Figure 1.

4. The Solution

The solution to this problem lies in capturing and maintaining the government knowledge as a central infrastructure, which could be used by diverse software applications. This government knowledge, consisting of government policies, translated into programmes and processes and their associated rules and regulations, which form the business logic of various e-Governance applications, should ideally be maintained by the policy makers themselves. In fact, the government knowledge, which is locked up in the policy maker's mind as tacit knowledge, would be made explicit and formal, thereby removing all ambiguity that is the failing of natural language depiction. In fact, this is the most important area of e-Governance and to which no attention has been paid by the government so far. Though efforts are under way to prepare data standards for the government, data standards by themselves will not facilitate the policy maker in modeling the government policy directly and unambiguously so that it can be re-used across software applications.

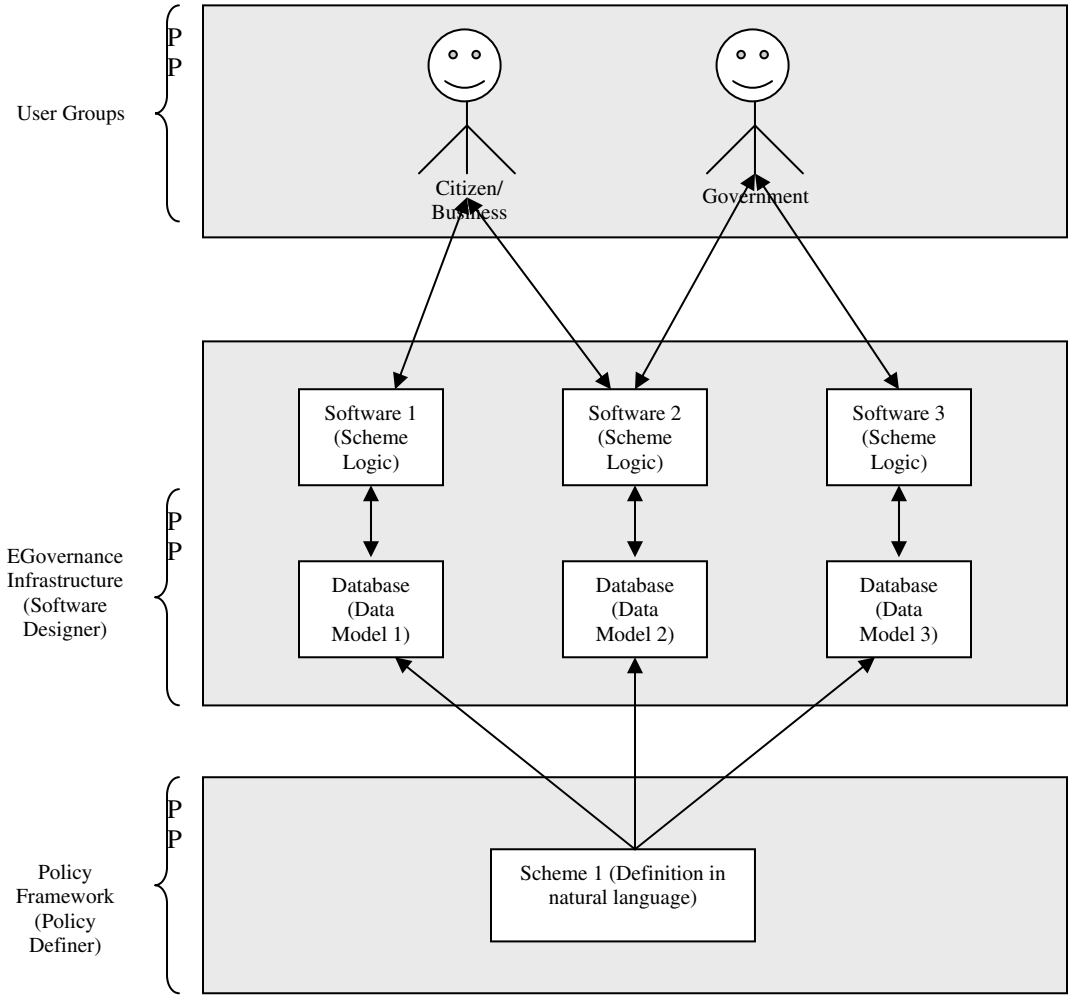

Figure 1: Current Architecture of E-Governance

In order to facilitate this aspect of e-Governance, the following would be required:

- A technology mechanism to efficiently model the government knowledge as a unified framework
- A set of tools and technologies to enable the policy maker to define and/or update the policy decisions of the government in a manner that is unambiguous and consumable and re-usable across software applications. For e.g. whenever a new scheme is launched or when any policy change is made to an existing scheme, then the policy maker himself should create and/or modify the details in the central government knowledge repository. This would ensure that the government knowledge is captured exactly as intended by the policy maker. Once this knowledge is captured, each software application that needs to use this knowledge can draw it directly from the central repository rather than hard coding it in its application.
- A technology solution to expose the government knowledge in a standard, platform independent manner so that the dependent software applications, irrespective of the environment in which they operate, are able to connect to the government knowledge infrastructure and consume it.

The following sections describe how each of these requirements could be met.

4.1 Modeling the Government Knowledge

In the past few years, the semantic web has gained lot of attention in view of the fact that the current state of web is not very conducive to intelligent search and retrieval of information. Tim Berners Lee, the inventor of the World Wide Web, envisioned a Semantic Web, which would be rich in the background knowledge about the various resources on the web and thus enable software agents on the web to interact intelligently and provide a range of automated services to people. The vision had a tremendous impact on the academia and the industry alike. A large number of research projects have been undertaken to develop tools and technologies to fulfill the vision. Notable among them have been the series of web standards proposed by W3C to represent the knowledge of any domain in the web in a standard manner. All research efforts have adopted *ontologies* as the underlying mechanism to represent the knowledge on the web.

5. Government Ontology to Represent Government Knowledge

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. In the field of computer science and artificial intelligence, an ontology is defined as a formal representation of a set of concepts within a domain and their interrelations. An ontology is used to define a domain and also to reason about the domain. It is a form of representation of knowledge about the domain of discourse. An ontology consists of concepts modeled as classes or sets, attributes of concepts modeled as properties, relations among the concepts and the instances of the concepts. Depending on the comprehensiveness of the language used to express an ontology, it may also include axioms, restrictions and rules which help in making assertions about the concepts in an ontology and also help in reasoning about the ontology and its associated knowledgebase. Thus, an ontology provides a common model for the stakeholders in a domain who need to share information in a domain. The fact that the ontologies are specified in a formal manner ensures that they are machine interpretable and can be used by software agents.

An ontology is very much akin to the way human beings store their knowledge and use it to communicate with other human beings. A software agent requires a similar mechanism to represent and reason about a domain and use it communicate with other software agents provided all of them commit to the same ontology, either directly or indirectly. In order to be able to perform the above tasks, an ontology needs to be represented in a formal manner, which can be used by machines to process the knowledge.

There are many reasons why an ontology of the government domain should be developed and maintained:

- It enables human users as well as software agents to share a common understanding of the

structure of information available in the government domain.

- It enables reuse of domain knowledge. Each software application only needs to provide the front-end to the user; the domain knowledge can be re-used by multiple software applications.
- Currently domain knowledge is mostly tacit, mostly in the heads of government functionaries or hardcoded in the software applications. Explicit specification of the knowledge makes it amenable for change and also makes it easy for any body to understand. This can becomes a mechanism for sharing best practices across governments.
- Separating the domain knowledge from operational knowledge is another advantage of building and maintaining an ontology. While the definition of a scheme in terms of its various properties would be a domain knowledge, the data relating to implementation of a scheme are the operational aspects which must be kept separate.

A number of languages are available to formally specify the ontologies. The most important among them is the OWL Web Ontology Language. As the name suggests, OWL is intended to provide a language that can be used to describe classes and relations between them that are inherent in Web documents and applications. OWL is web-standard endorsed by the W3C Consortium and has three *species* of languages in increasing order of their expressiveness viz., Owl-Lite, OWL-Dl and OWL-Full:

- **OWL-Lite** supports those users primarily needing a classification hierarchy and simple constraint features.
- **OWL DL** supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL was designed to support the existing Description Logic (hence the acronym DL) business segment and has desirable computational properties for reasoning systems.
- **OWL Full** is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.

Ontologies are becoming the order of the day for modeling any domain. The government domain is probably one of the most well structured domain and is highly amenable to ontological engineering. The first step towards beuiding government ontology would be to analyze each domain and build an ontology for each area in the government domain. For example, to continue with our own example, we could model the domain of developmental governance by capturing the knowledge of the domain as an ontology. It may be noted that it is not necessary that there should be a single ontology of the government. The ontologies could be distributed; however, they should be amenable to integration. For eg., there could be an ontology for the administrative structure of the government and another ontology for the programmes of the government. Alternatively, they could be maintained in a single ontology. The ultimate vision is that the users of these ontologies should view them as a single entity, be it a human user or a software agent.

6. Tools and Technologies for Managing the Government Ontology

Once we decide that the government knowledge could be modeled as ontologies, the next step is identify/create tools and technologies that could be used to manage the ontology of the government domain. There are many tools available to edit and maintain ontologies. Protégé developed by Stanford University is one such Open Source tool that is very well known and freely available in the public domain. However, these tools are very generic and could be intimidating to a policy definer who is not familiar with the technical terms used in these tools. What is required is a software tools that wraps such tools with a user interface which speaks the government language. There are three aspects to the management of the government ontology:

- An ontology editing system using which the policy maker models the government policy on the currently defined model. For eg., if a model of the Central Government Schemes is already

available, a policy definer in a central government department may simply create a new scheme as per the pre-defined ontological structure. This interface would be a much simpler interface using very government-specifc user interface.

- An ontology management system which would relate more to the administrative management of the ontology. Such a system would be required to faciliate structural changes to the government ontology. This interface would be more technical and may be managed by a technically oriented actor.
- The very fact that an ontology is subject to modifications entails that there should be a mechanism to maintain the versions of the ontology and relate the operational data to the correct versions of the ontology. For eg.,the plan approval process for a planning unit may be changed through a government notification. Such changes would of course be reflected in the ontology. Along with the changes in the ontology, the ontology management system should also have the facility to ensure that all data related to the older process flow are maintained and all the new process instances are related to the new approval process.

7. Exposing the Government Ontology

The third requirement is to expose the government ontology in a manner in which software applications can consume them, irrespective of the execution environment of the software. There are two aspects to this requirement:

The first aspect is to publicly make available the government knowledge in a manner which can be understood by the software developers so that they can consume them in their software applications. This requires that the government knowledge be exposed as small fine grained components thereby allowing software applications to use them. Secondly, these components should be such that they can be accessed over the web and consumed by any software application irrespective of the environment of the consuming application. Both these requirements are aptly fulfilled by Web Services. Web Services have supporting protocols for defining, publishing and discovery of services on the web. What is required is to enrich the government web services by adding a semantic dimension to it.

A government directory of web services, maintained by the government, would be required through which software developers would be able to view the list of available government web services and subscribe to the required services. If semantic web services are used, this directory could be a self-managing directory. The announcement of new services will be made through this directory. Any changes made to the government ontology and hence to the services would also be announced automatically by the directory. Some of the changes to the government ontology made by the policy makers would not require changes to the consuming software applications; the changes to the government ontology would be automatically reflected through the software applications. However, some changes to the government ontology could result in changes to the consuming applications. A mechanism has to be worked out to ensure that the concerned software managers are informed about the changes and also that the end user, who may be another government functionary or an external stakeholder such as a citizen or business, is also informed about the inconsistency in the software so that an inconsistent software is not used by the end users. It may be noted here that since the entire business logic is exposed as web services, these software applications would merely act as a front-end tool to expose the government service. This opens up a possibility for the software providers to make the software as attractive as possible to end users by adding value to the software. The citizen would also get a bouqet of service providers from which to choose. Once these requirements are fulfilled, the proposed architecture of e-Governance is shown in Figure 2.

8. Related Work

Some research work has already gone into this area. In particular, a number of EU-funded research

initiatives have been undertaken in this direction. These include, inter alia, the TERREGOV, OntoGov, AccesseGov, SemanticGov projects. All of these initiatives have used semantic technologies to create various centralized government infrastructure to achieve the different purposes of the projects. Each project has attempted to cover different aspects of governance and demonstrated their feasibility through pilot applications. Each of these research projects has demonstrated the capability of semantic technologies to overcome the problems highlighted above and have also developed a number of tools and technologies which could be put to good use. The State of the Art Report of the Access e-Gov project of the EU provides a brief description of each of these projects. A number of sample ontologies have been created under the research projects mentioned above. For e.g., the SemanticGov project provides the ontologies for the Governance Enterprise Architecture. The OntoGov project provides a comprehensive group of ontologies - the MetaOntology to descrine the concepts needed to describe services, Domain Ontology containing domain knowledge, Web Service Orchestration Ontology to enable dynamic composition of services and so on.

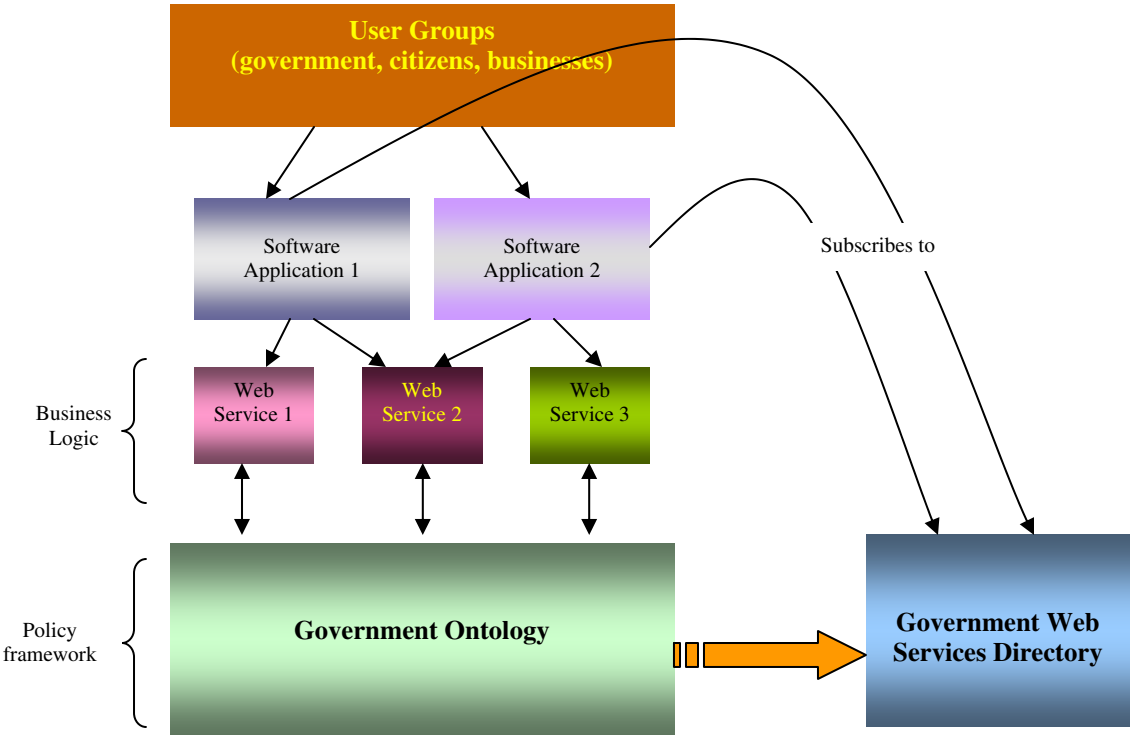

Figure 2: Proposed Architecture of E-Governance

9. Proposed Work in Indian Government Domain

Adopting the above approach, it is proposed to undertake a research project in the Indian Government domain. The area of development governance has been taken up as the domain for research. The development governance in the context of Indian Government is mainly constituted of schemes/programmes of the governments at various levels. A study of the same was made by the authors as part of the development of a software, PlanPlus, for planning in government. Based on the understanding of the domain developed during the project, an ontology of government schemes was developed using Protégé ontology development tool. A snapshot of the same is given in Figure 3.

The next step would be to develop a set of web services to access the knowledge encapsulated in the scheme ontology. PlanPlus would be taken up as a sample software application which will consume the web services exposing the government ontology. This would help in testing the proposed approach. Once tested, the next step would be to develop platforms for the management of the ontology and the development of a self-managing government web services directory. It is proposed to explore the platforms already developed under various research projects such as OntoGov, AccessEGov etc. and explore the possibility of using/customizing these platforms for the Indian Government domain.

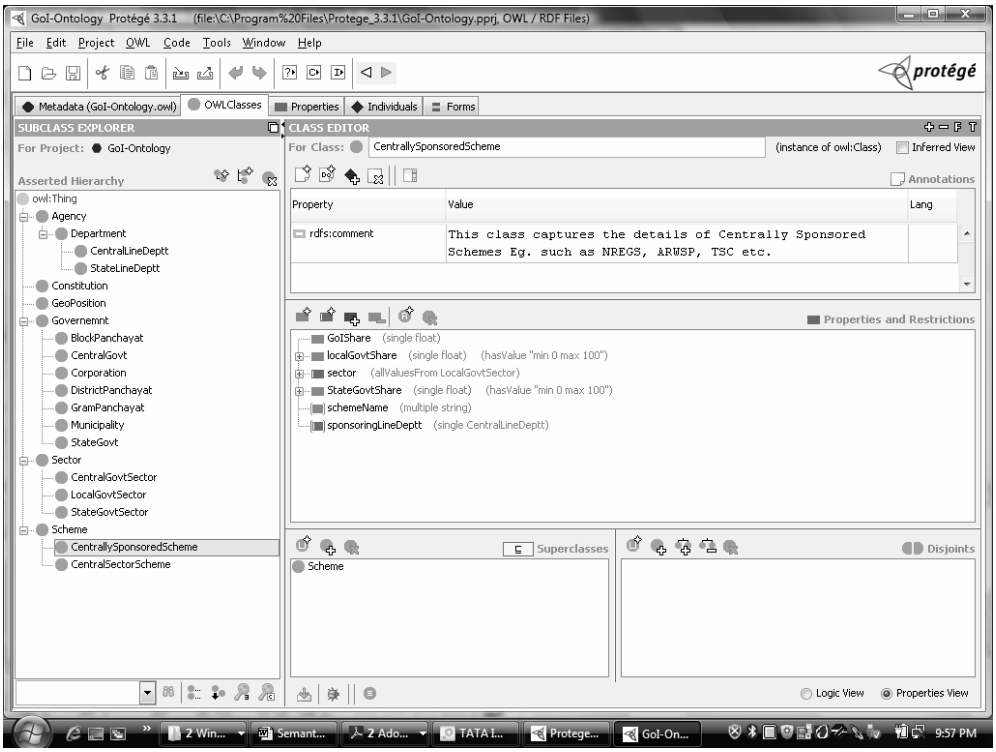

Figure 3: An Ontology of Government Schemes

10. Conclusion Remarks

The paper explored the approach of using ontologies to model the government knowledge. It is expected that this approach would cover the most important aspect of governance, viz., modeling the policies of the government directly by the policy makers. The approach explained here has been demonstrated through various research projects undertaken abroad. It is proposed to adopt this approach on a pilot basis for the Indian Government and after successful implementation of the pilot, would be scaled up to cover the entire gamut of Indian governance scenario.

References

1. Mittal, P.A., Kumar, M., Mohania, M K, Nair, M, Batra, N, Roy, N, Saronwala, A, and Yagnik L (2004) A framework for eGovernance Solutions, *IBM Journal for Research and Development*, Vol. 48, No. 5/6.
2. Uschold, M and Gruninger, M (1996), Principles, Methods and Applications, *Knowledge Engineering Review*, Vol. 11, No. 2.
3. http://www.w3.org/TR/2004/REC-owl-guide-20040210/ , accessed during May-July, 2008
4. http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_6_2_06_3_35_24_PM.pdf, D2.1 State-of-the-art Report. , accessed during May-July, 2008