Design and Development of Chemical Ontologies for Reaction Representation

Punnaivanam Sankar*,† and Gnanasekaran Aghila‡

Department of Chemistry and Department of Computer Science & Engineering and Information Technology, Pondicherry Engineering College, Pondicherry 605 014, India

Received December 15, 2005

This paper describes the development of chemical ontologies applied to the representation of organic chemical reactions. The ontologies are built using the methodology known as methontology. The hierarchically structured set of terms describing the subdomains, namely, organic reactions, organic compounds, and reagents, are constructed into individual ontologies. The ontologies consist of about 200 concepts and around 125 individuals. A set of binary relations is defined in order to integrate the ontologies with applications. The ontologies are implemented as an XML application with a set of vocabulary describing the domain knowledge. This paper also features an easy-to-use chemical ontological support system (COSS) intended to represent organic chemical reactions automatically. As a model application, the automatic representation of aliphatic nucleophilic substitution reactions is demonstrated using COSS. The paper also describes a keyword-based search system whose functionality is backed with COSS.

INTRODUCTION

Ontology is a formal specification of a shared conceptualization.[1] It provides a common vocabulary of an area in which the meaning of the terms and the relations between them are defined with different levels of formality. In the computer science domain, ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed upon understanding of the domain which may be reused and shared across applications and groups.[2] A chemical ontology tries to conceptualize the chemical knowledge in a narrow or broader perspective depending on the granularity level of formalization. In recent years, the importance of building chemical ontology has been felt in the context of evolving Web-based chemistry.[3] However, reports reveal that only a few domain-specific chemical ontologies are developed so far.[4−6] This may be attributed to the fact that the description of chemical knowledge is difficult because of the fuzziness involved in it, and building a chemical ontology requires considerable effort. Among the reported chemical ontologies, the ontology for the identification of functional groups and the semantic comparison of small molecules[4] describes an interesting application of detecting structurally similar compounds from chemical databases. In this work, the small molecule ontology is created on the basis of chemical functional groups automatically assigned by a computer program. The functional group terms generated are arranged suitably to obtain the ontology. The functional group taxonomy is utilized as an efficient pharmacophore search system. Another chemical ontology[5] describing the knowledge about the chemical elements in the periodic table is the first one created by a systematic methodology termed as methontology for the development of chemical ontologies. The methontology framework enables the construction of ontologies at the knowledge level, suggesting the main activities of the ontology development process as requirement specification, conceptualization, integration, implementation, and maintenance. In the above activities, the implementation stage of ontology development is very significant because it concerns the selection of a suitable knowledge representation language. A number of ontology specification languages evolved during the past decade. While the traditional languages such as Ontolingua,[7] LOOM,[8] OCML,[9] and so forth are already used for representing knowledge inside knowledge-based applications, recent Web-based ontology specification languages such as SHOE,[10] RDF,[11] RDF Schema,[11] XOL,[12] OML,[13] OIL,[14] and DAML+OIL[15,16] made an impact in the development of the Semantic Web. However, all of these languages are still in a development phase and are continuously evolving.

In developing a chemical ontology, the selection of a suitable language is crucial because the utility of the ontology mainly depends on the implementation language. Among the XML conforming languages, chemical markup language[17] (CML) has evolved as an interoperating XML-based markup language for describing the management of chemical information. The interoperability of CML with the emerging family of XML-based languages positions CML as a valuable tool in the context of developing Web-based chemistry.[18−20] Through a concise set of tags and their attributes, CML provides a base functionality for describing atomic, molecular, and crystallographic information. Subsequently, CMLReact,[21] an application of CML to describe chemical reactions, is slowly evolving. CMLReact shares its components with the CML components to manage chemical and biochemical reactions by including reaction components into the current CML functionality. The representation of reactions is not so straightforward when compared to the description of molecules. A simple structure description of molecules in a reaction is not sufficient to represent a complete chemical reaction. This is because the reaction

* Corresponding author e-mail: gapspec@yahoo.com.
† Department of Chemistry.
‡ Department of Computer Science & Engineering and Information Technology.

10.1021/ci050533x CCC: $33.50
Published on Web 09/20/2006

representation needs some more semantics. A molecule may behave as a reactant or a reagent or a main product or a byproduct or a solvent or even a catalyst depending on the nature of the reaction. Not only this, a chemical reaction has other components, such as arrows indicating the formation of products or the pathways, a "+" sign on both the reactant and product sides, electron movements normally shown by curly arrows, reversibility of a reaction, and so forth, which are difficult to encode. However, these difficulties can be sorted out by making use of markup methodologies and establishing relations between various components of a reaction. In fact, a chemical inferential work[22] to formalize chemical substantives such as structural formulas, compounds, and states emphasizes the significance of formalization of the language of relational chemistry (LRC). With a view to fully formalize LRC, the possible representations of substantive (nounlike) and relational (verblike) elements are manipulated to find a consistent formulation which allows some aspects of chemical behavior to be automatically modeled by the algebraic behavior of the formal description. In this work, the chemical relations such as "is isomeric with" and "reacts to form" are considered as some of the relational representations to arrive at information-rich substantives.

The existing systems to formalize reactions such as ChemDraw[23] and ISIS/Draw[24] use graphical-oriented formats. In contrast, SMILES,[25] SMIRKS,[26] REACTOR,[27] and so forth use the extensions of molecular formats stored as notations to represent chemical reactions. Markup technique is not the basis for this formalization. A pioneering work using the XML technology is CMLReact, which describes the chemical reactions using codes written in CML. Because the proposed ontologies are primarily developed to describe chemical reactions, the ontology implementation language as well as the reaction representation language must be one and the same to have complete interoperability with the other XML-conforming languages such as CML, CMLReact, SVG,[28] MathML,[29] and so forth. For these reasons, the implementation of the proposed ontologies is effected with XML syntax. It is an XML application with its own set of vocabulary. Though some of the terms are common with CMLReact, the present work is considered as a separate XML extension with its namespaces.[30] One of the important criteria in the development of a knowledge-based system in chemistry involves the building of reusable components. Thereby, the new system interoperates with the existing system in such a way that the declarative knowledge, problem solving techniques, and reasoning services can be shared among systems. This approach also enables the building of bigger and better systems inexpensively. XML applications facilitate the development of reusable components. In view of this, a custom XML application may be employed as a powerful ontology specification language in developing conceptual bases needed for building technology that allows knowledge reuse and sharing in the chemistry domain.

The main objective of the present work is to develop ontologies to serve as supporting frames for the representation of common organic reactions. Basically, an organic chemical reaction can be represented as [substrate + attacking reagent → (transition state) → product (main product + byproduct)]. Apart from these components, the conditions such as pressure, temperature, the presence of catalysts, and so forth also have a significant role in determining the nature of the products of the reaction and the reaction pathways. Except for the pressure and temperature conditions, the constituents of the other components are chemical molecules. Thus, the representation of a chemical reaction is essentially the representation of molecules which in turn is accomplished by atom-level and bond-level descriptions. CMLReact handles the reaction markup through the inclusion of molecular species such as reactants, products, and so forth in a container element ⟨reaction⟩. The description of structural features of the molecular species is handled with two different approaches. In the first approach, CMLReact allows the description of reaction markup including the complete description of each molecular species involved. The other approach allows the molecular information to be described as separate CML ⟨molecule⟩s assigned with a unique attribute "id" value and establishing the links in the reaction markup. According to this approach, the entire molecular markup need not be included in the reaction markup. Though this approach holds well for a succinct markup with enhanced human readability, it suffers from the fact that the mapping of atoms in the reactant to those in the product needs to be established through a set of links. This makes the reaction markup complicated.

In a chemical reaction, the nature of reacting species and reagent media play a crucial role in deciding the type of product and the mechanistic pathways through which the products are formed. The identification of correct products with respect to reaction conditions can be handled with chemical ontologies through the establishment of relationships between the reactant, reagent, and product. For example, the products of the reaction of an alkyl halide with moist silver oxide and dry silver oxide are aliphatic monohydric alcohol and aliphatic ether, respectively. Suitable ontological relationships between the concepts alkyl halide, moist silver oxide, dry silver oxide, aliphatic monohydric alcohol, and aliphatic ether help in fixing the correct product. Similarly, by defining relationships for aqueous potassium hydroxide with a nucleophilic substitution reaction and alcoholic potassium hydroxide with an elimination reaction, the correct reaction selection can be achieved. The knowledge-level ontological descriptions impose an artificial intelligence perspective in the reaction representation. Thereby, the need for the specification of product-side information can be eliminated while deriving applications. This conveys the meaning that the reaction markup, when supported with needed knowledge specific to chemical reactions, makes the system intelligent enough to capture the correct reaction paths and exact products automatically. The way of achieving this is to support the markup with ontologies representing formal descriptions of terms, concepts, behavior, relationships, and so forth in the domains related to organic reactions. In fact, the relevance of ontologies in chemistry and tasks for the chemical community is underlined in the early development of CML, in which the chemical reaction is mentioned as one of the chemical-specific information types which need ontological support.[3]

The creation of ontologies describing all of the knowledge related to organic reactions needs the complete definitions of a wide range of organic compounds and the reagents associated with the reactions, making the process very tedious and time-consuming. In the present study, a general design

methodology is proposed with a restricted number of individuals which are necessary and sufficient to make the system as a representative model. In the reaction classification, a complete description of a specific reaction type, namely, the aliphatic nucleophilic substitution reaction, is only attempted. As a model application, an automatic representation system for the aliphatic nucleophilic substitution reaction is developed. Subsequently, the utility of the ontologies in framing a powerful semantic search system is also demonstrated. Though the present system is restricted to only primitive reactions, the proposed strategy can hopefully be employed to develop or extend the ontologies for other types of reactions. It is envisioned to create a powerful knowledge base of chemical reactions by integrating various ontologies related to this domain. Thereby, the resulting system can be used to develop applications in both academic and research areas. The possible applications on the academic side are the development of expert systems behaving as intelligent tutoring and authoring agents. On the research side, it is envisaged to develop ontologies describing the molecular structures with atom-level and bond-level granularity. This development, in combination with the reaction ontologies, is expected to form a foundation for the development of useful tools to share and reuse the knowledge repository in supporting the process of drug design and drug discovery.

ONTOLOGY DEVELOPMENT

Methodology. The chemical ontologies are constructed following the methodology described by methontology. As there is no fully matured methodology for ontology construction, the selection of this methodology is made by comparing the existing methodologies with respect to the proposed ontologies in the chemical domain. The overview of methodologies[31] for building ontologies reported earlier provides a diagnosis of the state of the art of methodologies for ontology development through the analysis of best known methodologies against the IEEE Standard[32] for Developing Software Life Cycle Processes 1074−1995. The report prescribes that the methontology approach is the most matured and most consensual in the field. Further, it is recommended by The Foundation for Intelligent Physical Agents[33] (FIPA) for ontology construction. FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. A comparative study of most representative methodologies to build ontologies from scratch reveals that one of the important criteria for analyzing the methodologies is the strategy for building ontologies. This deals with the dependency of the ontology developed and its final application. Accordingly, the methodologies are classified into application-dependent (KACTUS project), application-semidependent (SENSUS-based methodology and Gruninger and Fox's methodology), and application-independent (Cyc, Uschold and King, and methontology) approaches.[31] In the application-dependent method, the ontology is built on the basis of an application knowledge base by means of a process of abstraction. In the case of the application-semidependent methodology, possible scenarios of ontology use are identified in the specification stage. In contrast, in the application-independent approaches, the process is totally independent of the uses to which the ontology will be put in knowledge-based systems, agents, and so forth. The present study is aimed at the development of chemical ontologies at the knowledge level to derive many applications through the reuse of ontologies. This makes the choice of selecting any one of the application-independent methodologies.

Further, the maturity of the methodology is measured with criteria like the recommended life cycle, the difference between the methodology and IEEE 1074−1995, recommended techniques, and the ontologies developed using the methodology along with the systems built using the ontologies, of which, the ontology life cycle identifies the activities on various stages of the development of the ontology. On the basis of this criterion, methodologies are analyzed whether they propose any implicit or explicit life cycle. This makes the selection of methontology a suitable methodology for the present work because this approach identifies the sets of stages through which the ontology moves during its lifetime, describing what activities are to be performed in each stage and how the stages are related. It proposes the activities in such a way to build the ontologies with some assurance and completeness even by geographically distant cooperative teams. The ontology life cycle of the methontology approach is an evolving prototyping life cycle composed of a series of development-oriented activities such as requirement specification, conceptualization of the domain knowledge, formalization of the conceptual model in a formal language, implementation of the formal model, and maintenance of the implemented ontologies. Along with these main activities, the methodology proposes some support activities such as knowledge acquisition, documentation, evaluation, and integration of other ontologies to be performed along the whole ontology building process. So, the process prescribed by the methontology approach is followed to create chemical ontologies.

According to the methontology approach, the design criteria include the following phases: requirement specification, conceptualization, integration, and implementation. The requirement specification is built after acquiring the domain knowledge through reference books.[34−37] The specifications include the purpose, the level of formality, and the scope of the ontologies to be developed. The main purpose of the proposed ontologies in the domain of organic chemical reactions is to use them as a supporting system to develop a model to represent organic reactions automatically. However, the knowledge can be reused for deriving advanced applications and tools in the fields of organic analysis, drug discovery, drug design, and so forth, by incorporating additional definitions and imposing restrictions on the concepts in terms of axioms. The level of formality is fixed as semiformal. The ontology is implemented in XML with a vocabulary set representing the domain knowledge. The scope of the ontologies includes a list of about 200 concepts and around 125 individuals collectively in the domain of organic reactions, organic compounds, and reagents in its initial stage. Further, this range can be widened by including more definitions

Conceptualization. In the conceptualization phase, the acquired knowledge in an unstructured form is organized into structured intermediate representations using external representations which are independent of the implementing language. This converts the informally perceived view of

the domain into semiformal intermediate specifications. This phase includes the tasks of constructing a glossary of terms; building concept hierarchies; preparing a data dictionary; and constructing the instance tables and attribute tables, binary relation diagrams, and axioms. The vocabulary describing concept classes, individual instances, and their attributes in the domain of chemical reactions are collected and organized in the form of a glossary of terms. Attempts on classifying the concepts into related groups resulted in three major classifications covering the three subdomains, namely, the classification of organic reactions, organic compounds, and reagents. These classifications are structured into hierarchical concept taxonomy within the subdomains using the conventional relations such as *subclassOf, subclassPartitionOf,* and *exhaustiveSubclassOf*.

The *subclassOf* relation is used to relate the concepts having a parent−child relationship resembling the usage of the traditional *is-a* relation. For example, in the classification of reaction classes, the concepts "addition reaction" and "substitution reaction" are related to the concept "organic reaction" using the *subclassOf* relation. This allows that every instance of addition reaction and substitution reaction is also an instance of the parent concept, organic reaction. In contrast, the relation *subclassPartitionOf* relates the parent concept with a set of child concepts which are mutually disjointed. Electrophilic substitution reaction, nucleophilic substitution reaction, and free radical substitution reaction are a set of subclasses related to the same concept "substitution reaction". However, these concepts are mutually disjointed. It means that the instances of one concept cannot be an instance of the other. Similarly, aliphatic nucleophilic substitution reaction and aromatic nucleophilic substitution reaction are mutually disjointed classes related to the parent concept "nucleophilic substitution reaction" with the *subclassPartitionOf* relation. An *exhaustiveSubclassOf* relation relates a concept to a set of mutually disjointed subclass partitions covering the entire concept. Every instance of the parent concept is an instance of exactly one of the subclasses in the partition. For example, the concept "aliphatic hydrocarbon" has a set of mutually disjointed subclasses, namely, alkane, alkene, and alkyne. The subclass partition is exhaustive, and no further partition is possible for the parent concept. Any instance of aliphatic hydrocarbon is an instance of only one of the subclass partitions. The classifications of vocabulary identified with proper relations in the domain of interest produced three distinct ontologies as prescribed by methontology. The ontologies are identified as organic reactions ontology, organic compounds ontology, and reagents ontology.

The concept classifications for the chemical reactions are shown in Figure 1. In the reactions ontology, the upper-level classification starts with the primary classes such as addition reaction, substitution reaction, and so forth. Further branching of the concepts depends on the subclasses of the concepts and their hierarchical relationships. Classification in the bottom level ends with the concepts holding individual instances. In the classification of substitution reaction, the concept "substitution reaction" is classified into three subclasses, namely, free radical substitution, electrophilic substitution reaction, and nucleophilic substitution reaction. The electrophilic and nucleophilic substitution reactions are further classified into their corresponding aliphatic and aromatic reactions. The concept tree branching ends with the node containing individuals shown as "reactionInstance", indicating the individual reactions are attached to this concept. The individual instances are created by attaching values to the attributes defined for the concept holding the individuals. For example, "reaction of alkyl halide with aqueous potassium hydroxide" is an instance reaction of aliphatic nucleophilic substitution. So, this instance can be created during implementation by attaching this name as the value to the title attribute for the concept. Similarly, a number of such individual reactions of the type can be defined as the instances simply by providing new reaction names belonging to this class as the values of the title attribute. The identity of a reaction instance with respect to the respective parent classes is tied with the specific values associated with the id attributes.

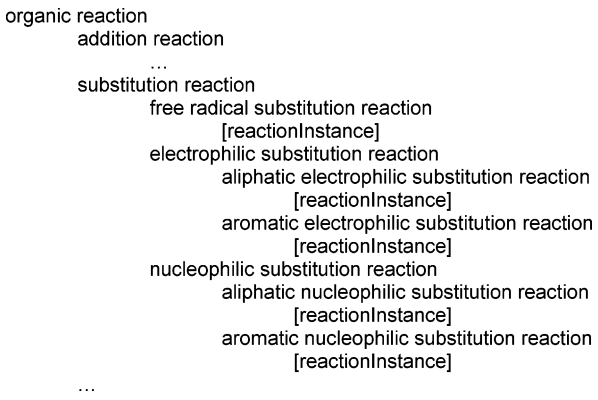

Figure 1. Ontological classification of the substitution reaction.

The classification of organic compounds includes the common class of organic compounds. About 20 classes of organic compounds are considered to cover almost all of the common classes of organic compounds. However, in defining the individual members of each class, only five to six individual definitions are considered for the present work. The inclusion and study of many individual compounds stretches beyond the scope of the present work. The restriction of defining only five to six individuals in each class of organic compounds itself accounts for about 125 definitions for individual common organic compounds with which the common reactions can be described. A portion of various classes of organic compounds in their hierarchical structure is shown in Figure 2. In this classification, the individual instance holder is "molecule".

The third subdomain, reagents, includes both organic and inorganic reagents categorized into two main groups, nucleophilic and electrophilic reagents. The concept hierarchy for nucleophilic reagent includes possible nodes such as negative nucleophilic reagent and neutral nucleophilic reagent. The term "reagentInstance" is used to hold individual reagents. Common nucleophilic reagents of both organic and inorganic nature such as aqueous potassium hydroxide, sodium methoxide, silver acetate, sodium acetylide, aqueous ethanol, and so forth are some examples for individuals defined under the nucleophilic reagent class. This classification includes about 24 reagents commonly employed in aliphatic nucleophilic substitution reactions. The concept hierarchy showing the taxonomy of nucleophilic reagents is presented in Figure 3.

```
organic compound
    …
    halogen compound
        aliphatic halide
            aliphatic alkyl halide
                mono halogen alkyl halide
                    [molecule]
                di halogen alkyl halide
                    [molecule]
                tri halogen alkyl halide
                    [molecule]
                tetra halogen alkyl halide
                    [molecule]
            aliphatic alkenyl halide
                …
            aliphatic alkynyl halide
                …
        aromatic halide
            …
    alcohol
        aliphatic alcohol
            aliphatic monohydric alcohol
                [molecule]
            aliphatic dihydric alcohol
                [molecule]
            aliphatic trihydric alcohol
                [molecule]
        aromatic alcohol
            …
    …
```

Figure 2. Taxonomy of a few organic compound classes.

```
reagent
    electrophilic reagent
        …
    nucleophilic reagent
        negative nucleophilic reagent
            [reagentInstance]
        neutral nucleophilic reagent
            [reagentInstance]
```

Figure 3. Concept classification tree for nucleophilic reagents.

Binary Relations. The next step in conceptualization is building binary relation diagrams to establish the relationship between concepts of the same as well as different ontologies. This results in a dynamic linking of concepts which are useful in setting the guidelines for integrating ontologies. In the present work, the concepts relating to the various components of the reaction are defined into three major classifications. To establish the relations between these concepts belonging to different taxonomies, suitable binary relations are identified. Accordingly, a concept "aliphatic nucleophilic substitution reaction" belonging to the reaction ontology can be linked to a concept "nucleophilic reagent" classified in the reagents ontology. Thereby, the instances of the nucleophilic reagent class are included into the nucleophilic substitution reaction. Similarly, the concept "reaction of alkyl halide with aqueous potassium hydroxide" related with the concept "ethyl bromide" includes the specified individual as a component of the reaction. In the same manner, the product components of a reaction can be linked to the corresponding reaction with suitable binary relations. A thorough analysis of the various concepts belonging to the three ontologies with a perspective of describing common organic reactions resulted in the identification of few important binary relations. Whenever a binary relation is defined between two concepts, it includes an inverse relationship too. The identified relations are named as *hasReactantComponent*, *hasReagentComponent*, *hasProductComponent*, *hasReactantClass*, *hasReagentClass*, and *hasProductClass*, for which the inverse relations are defined as *reactantComponentIn*, *reagentComponentIn*, *productComponentIn*, *reactantClassIn*, *reagentClassIn*, and *productClassIn*, respectively. When these relations are applied to various concepts, a number of binary relations are defined.

Individuals and Attributes. Three common attributes, namely, "id", "type", and "title", are defined in default for almost all of the concepts including individuals irrespective of the ontologies. The "id" attribute is used to provide a unique identification string value with which the concept can be identified. The string value for each "id" attribute is defined in such a way as to hold the information related to the hierarchy to which the concept belongs. For example, the value of "id" for the concept "reaction" is assigned as "sub/nuc/ali-001". The serial number at the end of the string makes the individual instance unique. The remaining part of the string "sub/nuc/ali" indicates the parent class to which the instance belongs. Each substring separated by "/" denotes one hierarchy level. The string values of the id attribute when terminated with "-ind" implies that the concept is holding a list of individuals. In the individual's description, the same is replaced with a serial number. Thereby, the above "id" indicates the individual with serial number "001" belongs to the class "aliphatic nucleophilic substitution reaction" whose parent class is nucleophilic substitution reaction which in turn belongs to the class substitution reaction as described in the following code fragment. Similarly for the individuals of compound class, aliphatic monohydric alcohol the 'id' value assigned is "alc/ali/mno-001" this indicates the hierarchy as follows:

```
substitutionReaction id = "sub"
    nucleophilicSubstitutionReaction id = "sub/nuc"
        aliphaticNucleophilicSubstitutionReaction id = "sub/nuc/ali-ind"
            [reactionInstance  id = "sub/nuc/ali-001"
                title=" reaction of alkyl halides with aqueous potassium
                    hydroxide"]
alcohol id = "alc"
    aliphaticAlcohol id = "alc/ali"
        aliphaticMonoHydricAlcohol id="alc/ali/mno-ind"
            [molecule id="alc/ali/mno-001"
                title="methyl alcohol"]
```

The "type" attribute is used to hold the meaning of the concept in several cases. For the individual "ethyl bromide", the type attribute is assigned a value of "alkyl halide", indicating that the compound belongs to the alkyl halide group. In reagents, for the individual "dry silver oxide" the value for the "type" attribute is "nucleophilic", indicating that the reagent is a nucleophilic reagent. The third attribute "title" is used for assigning the name values of the instance in all of the concept classifications. Apart from these three attributes, several attributes are identified for the instances of compounds and reagents. Considering the present perspective of the ontology to be used as a supporting tool for the representation of the reaction, the attributes which are necessary and sufficient to serve the purpose alone are included in the attribute list.

IMPLEMENTATION

The implementation stage involves the transformation of the conceptual model into an implemented model in which the intermediate representations are translated into formal machine-readable specifications using ontology representation language. The ontology specification is coded as an extension of XML. An ontology language must describe the object classes including the individual instances and the

relations between objects in order to capture the terms and their meanings. When the standards of XML are followed, the representations of concept classes, hierarchy of concept classes, binary relations, and individuals are described as both machine- and human-understandable codes and stored as XML documents. These XML documents are validated against a document type definition (DTD) in which the vocabulary needed to formalize the ontologies is declared according to the specifications. The code for the implemented conceptualization of the domain resulted in three basic codification patterns. The first type of code describes the hierarchical classification of concept classes excluding the instances. The second type describes the instances of the ontologies. The third one is used to describe the relations between the concepts. Because XML allows the nesting of elements implying the hierarchical layer of elements, the concept taxonomies are easily coded as nested elements within an XML document. This enables the query processing or a search process with a node-based search within the XML document. This representation in XML files allows the design of a mechanism to explicitly represent services and processes and to build models to extract the information.

Description of Classes. In describing the concept classes taking into consideration human readability, each concept identified as a class is considered as such as an XML element. These elements are nested according to the concept hierarchy. Accordingly, every node within a hierarchical structure becomes a unique XML element which requires a declaration in the DTD. This informs that, whenever a new concept class is defined within any concept taxonomy, the concept must be declared as an XML element along with the corresponding descriptions in the DTD for the validation. The sample code for the XML implementation of reaction classes, compound classes, and the reagent classes are presented in Figure 4. The nesting of elements is according to the hierarchy of the concepts appearing in the corresponding taxonomy. By comparing the concept classifications in Figures 1−3 with the respective XML code in Figure 4, it is clear that the patterns resemble each other, except for the fact that the latter is machine-readable. In addition to the description of concept classes, the implementation procedure includes the incorporation of needed semantics in the form of attributes and their associated values. Thus, the concepts are described along with the id and title attributes holding appropriate values. The annotations, if any, for the concept classes are made with the element <definition>*string*</definition>.

Description of Individuals. The second type of coding is needed for the description of individuals. The coding structure for an individual is different from that of the code describing the concept classes. This is due to the fact that the components or the parts of an individual to be described are not related with hierarchical relationships. The XML implementation of instances of reactions, compounds, and reagents is shown in Figure 5. The description of individual reactions is made using <reactionInstance> elements nested within the corresponding reaction class elements along with the appropriate attributes and their values.

As the present study is concerned with the representation of chemical reactions, the granularity level is fixed in such a way that the atom-level and bond-level descriptions are avoided. However, a molecule is considered as a combination of several chemical moieties. Because the individuals of aliphatic organic compounds having a single functional group are only considered for the ontology development, the compounds are considered as composed of a nonfunctional group bonded with a functional group. The nonfunctional group is composed of three groups bonded to a central carbon atom. With this skeleton, a common codification pattern is arrived at for the description of individual aliphatic compounds. According to this procedure, the inner structural features such as the functional group, nonreactive part, carbon center, and the atom groups attached to the carbon center are considered as XML elements, namely, <functionalGroup>, <nonfunctionalGroup>, <groupCentre>, and <group>, respectively. These elements are suitably nested into a <structuralFormula> element, which in turn is placed inside the <molecule> element. Along with the id and title attributes, the type attribute is included in the descriptions of individual compounds to provide additional meaning to the concepts. For example, the element <groupCentre> with a value of "C" for the formula attribute and holding a value of "sp3" for the type attribute implies that the group center is a carbon center which is sp3-hybridized. Similarly, in <functionalGroup> elements, the semantics such as halide, aldehyde, ketone, and so forth can be introduced with the type attribute.

In the case of the description of reagents, the above coding pattern does not match with that of an individual reagent molecule. This is because the components of a reagent molecule vary from the components of an organic molecule with respect to the role in the chemical reactions. Accordingly, two important components of any nucleophilic reagent considered for coding a nucleophilic reagent are a nucleophilic unit and a non-nucleophilic unit. These two terms are defined as XML elements and included in a container element named <nucleophile>, which in turn is placed inside the <reagentInstance> element. In addition to this, <medium> and <composition> elements are included in the markup to specify the aqueous and nonaqueous natures of the reagent medium and its composition. The <nucleophilicUnit> and <non-nucleophilicUnit> elements are associated with additional attributes such as count, charge, chargeCount, formula, and nucleophilicity. The values specified for these attributes are used in fixing the stoichiometry of the reaction at present. However, the importance of these attributes becomes significant while dealing with the determination of reaction mechanisms.

Description of Relations. A very important part of the implementation is encoding the relationships between concept classes and individuals belonging to the same or different concept taxonomies. The tags proposed for this purpose are <binaryRelation> and <binaryRelationInstance>. The binary relations play a crucial role during the extraction of knowledge or when integrating the ontology with applications. These relations are mainly used to relate concepts belonging to different concept taxonomies. Various components of binary relations are described within the element <binaryRelationInstance> with appropriate attributes being "id", "title", "source", "target", and "inverse". The name of the relation defined is the value of the attribute "title". A binary relation relates two concepts belonging to the different concept trees. These two concepts are identified as the source concept and a target concept whose names are assigned as

```
<!--reaction classification -->

<?xml version="1.0"?>
<!DOCTYPE organicReaction SYSTEM "rxnontoDTD.dtd" >
<organicReaction xmlns = “http://www40.brinkster.com/sankarpec/onto/”  id="orc"
        title="Reaction">
<!--substitution reaction-->
        <substitutionReaction id="sub" title="substitution Reaction">
                <definition></definition>
                <nucleophilicSubstitutionReaction id="sub/nuc"
                        title="Nucleophilic Substitution Reaction">
                        <definition></definition>
                        <aliphaticNucleophilicSubstitutionReaction id="sub/nuc/ali-ind"
                                title="Aliphatic Nucleophilic Substitution Reaction">
                                <definition></definition>
                                <reactionInstance id="" title="" />
                        </aliphaticNucleophilicSubstitutionReaction>
                </nucleophilicSubstitutionReaction>
        </substitutionReaction>
<!--contd-->
</organicReaction>

<!--compound classification-->

<?xml version="1.0"?>
<!DOCTYPE organicCompound SYSTEM "rxnontoDTD.dtd" >
<organicCompound xmlns = “http://www40.brinkster.com/sankarpec/onto/”
         id="org" title="organic compound">
        <definition></definition>
<!--halogen compounds-->
        <organicHalide id="org/hal" title="halogen compound">
                <definition></definition>
                <aliphaticHalide id="org/hal/ali" title="aliphatic halogen compound">
                        <definition></definition>
                        <aliphaticAlkylHalide id="org/hal/ali/ane" title="aliphatic alkyl halide">
                                <definition></definition>
                                <monoHalogenAlkylHalide id="org/hal/ali/ane/mno-ind"
                                        title="mono halogen alkyl halide" type="">
                                        <definition></definition>
                                        <molecule id="" title="" type="" formula="" IUPAC="" />
                                </monoHalogenAlkylHalide>
                        </aliphaticAlkylHalide>
                </aliphaticHalide>
        </organicHalide>
<!--contd-->
</organicCompound

<!--reagent classification-->

<?xml version="1.0"?>
<!DOCTYPE reagent SYSTEM "rxnontoDTD.dtd" >
<reagent xmlns = “http://www40.brinkster.com/sankarpec/onto/” id="rgt" title="Reagent" type="">
<!--nucleophilic reagents-->
        <nucleophilicReagent id="rgt/nuc" title="Nucleophilic Reagent">
                <negativeNucleophilicReagent id="rgt/nuc/neg-ind"
                        title="Negative Nucleophilic Reagent">
                                <reagentInstance id="" title="" type="" formula=""/>
                        </negativeNucleophilicReagent>
                        <neutralNucleophilicReagent id="rgt/nuc/neu-ind"
                                title="Neutral Nucleophilic Reagents">
                                <reagentInstance id="" title="" type="" formula=""/>
                        </neutralNucleophilicReagent>
        </nucleophilicReagent>
<!--contd-->
</reagent>
```

Figure 4. Sample XML document fragments describing the hierarchical classification of concepts from reaction, compound, and reagent ontologies.

the values of the attributes “source” and “target”, respectively. Also, a binary relation is always associated with an inverse relation for which the source and target concepts are reversed. The inverse relation name defined is assigned as the attribute value of “inverse”. In the present work, three different classifications describing reactions, reagents, and compounds need to be related with each other using suitable relationships. There is a set of binary relations, namely, *hasReactantClass*, *hasReagentClass*, *hasProductClass*, *hasReactantComponent*, *hasReagentComponent*, and *hasProductComponent*, for which the inverse relations are defined as *reactantClassIn*, *reagentClassIn*, *productClassIn*, *reactantComponentIn*, *reagentComponentIn*, and *productComponentIn*, respectively. All of these binary relations are defined to relate the components of reaction ontology with those of compounds ontology and reagents ontology. A specific reaction class is related to the corresponding reagent class in reagent ontology with the *hasReagentClass* relation along with the inverse relation *reagentClassIn*. In relating compound classes with reaction classes, the relations *has-*

```
<?xml version="1.0"?>
<!DOCTYPE aliphaticNucleophilicSubstitutionReaction SYSTEM "rxnontoDTD.dtd">
<aliphaticNucleophilicSubstitutionReaction xmlns = “http://www40.brinkster.com/sankarpec/onto/”
        id="sub/nuc/ali-ind" title="Aliphatic Nucleophilic Substitution Reaction">
        <definition/>
        <reactionInstance id="sub/nuc/ali-001"
                title="reaction of alkyl halide with aqueous potassium hydroxide"/>
        <reactionInstance id="sub/nuc/ali-002"
                title="reaction of alkyl halide with moist silver oxide"/>
        <reactionInstance id="sub/nuc/ali-003"
                title="reaction of alkyl halide with dry silver oxide"/>
        <reactionInstance id="sub/nuc/ali-004"
                title="reaction of alkyl halide with sodium methoxide"/>
        <!-- contd -->
</aliphaticNucleophilicSubstitutionReaction >
```

```
<?xml version="1.0"?>
<!DOCTYPE monoHalogenAlkylHalide SYSTEM "rxnontoDTD.dtd" >
<monoHalogenAlkylHalide xmlns = “http://www40.brinkster.com/sankarpec/onto/”
        id="org/hal/ali/ane/mno-ind" title="mono halogen alkyl halide" type="alkyl halide">
 <molecule id="m1" type="aliphatic" title="methyl chloride" formula="CH3Cl"
        IUPAC="Chloromethane">
        <structuralFormula>
                <groupCentre id="m1cc1" type="sp3" formula="C">
                        <group id="m1cc1g1" type="atom" title="hydrogen" formula="H">
                                <relation title="bond">
                                        <relationTarget id="m1cc1" title="groupCentre"/>
                                         <bond order="1" type="wedge" plane="above"/>
                                </relation>
                        </group>
                        <group id="m1cc1g2" type="atom" title="hydrogen" formula="H">
                                <relation title="bond">
                                        <relationTarget id="m1cc1" title="groupCentre"/>
                                        <bond order="1" type="wedge" plane="above"/>
                                </relation>
                        </group>
                        <group id="m1cc1g3" type="atom" title="hydrogen" formula="H">
                                <relation title="bond">
                                        <relationTarget id="m1cc1" title="groupCentre"/>
                                        <bond order="1" type="hatch" plane="below"/>
                                </relation>
                        </group>
                        <group id="m1cc1g4" type="halide" title="chloride" formula="Cl">
                                <relation title="bond">
                                        <relationTarget id="m1cc1" title="groupCentre"/>
                                        <bond order="1" type="flat" plane="in"/>
                                </relation>
                        </group>
                </groupCentre>
                <functionalGroup id="chloride" type="halide" title="chloride" formula="Cl"/>
                <non-functionalGroup id="Me" type="methyl" title="methyl" formula="CH3"/>
        </structuralFormula>
 </molecule>
 <!-- contd -->
```

```
<?xml version="1.0"?>
<!DOCTYPE negativeNucleophilicReagent SYSTEM "rxnontoDTD.dtd" >
<negativeNucleophilicReagent xmlns = “http://www40.brinkster.com/sankarpec/onto/”
        id="rgt/nuc/neg-ind" title="Negative Nucleophilic Reagent">
        <reagentInstance id="rgt/nuc/neg-001" type="negative"
                title="aqueous potassium hydroxide" formula="">
                <nucleophile id="" title="potassium hydroxide" type="" formula="KOH">
                        <nucleophilicUnit id="HO" type="anion" title="hydroxide" formula="OH"
                                count="1" charge="-" chargeCount="1" nucleophilicity="high"/>
                        <non-nucleophilicUnit id="" title="potassium" formula="K" count="1"
                                charge="+" chargeCount="1"/>
                </nucleophile>
                <medium>aqueous</medium>
                <composition>none</composition>
        </reagentInstance>
        <!-- contd -->
</negativeNucleophilicReagent >
```

Figure 5. Sample XML document fragments describing one individual instance from three ontologies.

ReactantClass and *hasProductClass* are used with their inverse relations *reactantClassIn* and *productClassIn*. These relations hold the semantics of whether a particular compound class plays the role of a reactant or a product in a specific reaction. In the same manner, individuals of a reaction type are related with the corresponding individuals in a compound class or reagent class with the binary relations *hasReagentComponent*, *hasReactantComponent*, and *hasProductComponent*, for which the inverse relations are *reagentComponentIn*, *reactantComponentIn*, and *productCompo-*

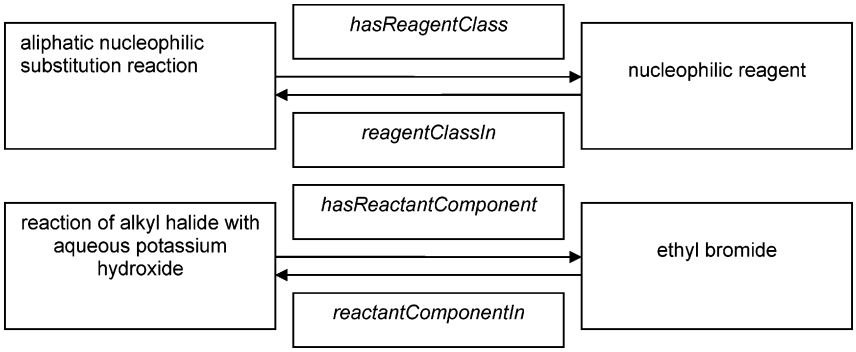

Figure 6. A representative binary relation diagram.

```
<binaryRelation>
        <binaryRelationInstance id="1" title = "hasReagentClass"
                source ="aliphaticNucleophilicSubstitutionReaction"
                target ="nucleophilicReagent"
                inverse ="reagentClassIn"/>
        <binaryRelationInstance id="2" title = "hasReactantComponent"
                source = "reaction of alkyl halide with aqueous potassium hydroxide"
                target = "ethylBromide"
                inverse = "reactantComponentIn"/>
</binaryRelation>
```

Figure 7. XML code fragment describing binary relations.

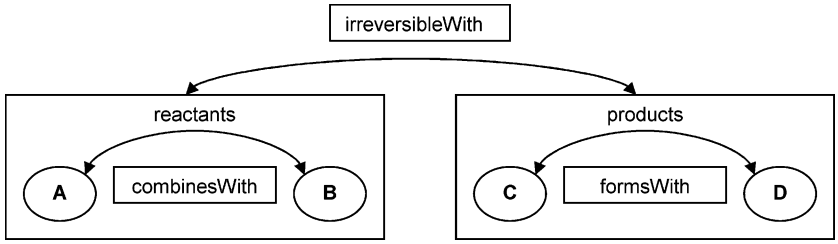

Figure 8. Components of a chemical reaction are shown using relationships.

nentIn. Representative binary relations specifying the relation name, source concept name, target concept name, and corresponding inverse relation name are presented in Figure 6 followed by the sample code describing them in Figure 7.

The other type of relationship is to establish relations between concepts which are not part of any concept hierarchy. In the representation of a chemical reaction, the encoding of the basic structure of a reaction is also important. For example, in the chemical reaction A + B → C + D, the meaning of the symbol "+" is different on both sides of the forward arrow. On the left-hand side, it means the reactants are combining, and on the right-hand side, it means the products are formed separately. Also, the products are shown to be formed from the reactants using a forward arrow. In the case of a reversible reaction, it is replaced by a two-way arrow. In the present work, an attempt is made to encode the semantics of this basic structure using the <relation> element. Accordingly, the components of a general chemical reaction are related to each other with suitable relations as shown in Figure 8. The XML code describing these relationships is given in Figure 9.

In the code given in Figure 9, the <relation> element is used to represent the general format of a chemical reaction. Accordingly, the components of a reaction are tied with this element. Also, it is used to indicate the reversible nature of a reaction. The structure of the <relation> element includes a <relationTarget> element. Both of these elements are associated with a title attribute. The title attribute for the <relation> element indicates the type of relation. In contrast, the same in <relationTarget> specifies the target element with which the relation is established. For example, the

```
<reaction>
        <reactantList>
                <relation title="irreversibleWith">
                        <relationTarget id="" title="productList" />
                </relation>
                <reactant>
                        <relation title="combinesWith">
                                <relationTarget id="" title="reagent" />
                        </relation>
                </reactant>
                <reagent >
                        <relation title="combinesWith">
                                <relationTarget  title="reactant" />
                        </relation>
                </reagent>
        </reactantList>
        <productList>
                <relation title="irreversibleWith">
                        <relationTarget id="" title="reactantList" />
                </relation>
                <mainProduct>
                        <relation title="formsWith">
                                <relationTarget  title="byproduct" />
                        </relation>
                 </mainProduct>
                <byProduct>
                        <relation title="formsWith">
                                <relationTarget  title="mainProduct" />
                        </relation>
                </byProduct>
        </productList>
</reaction>
```

Figure 9. XML coding pattern describing the relationships in a general chemical reaction.

```
<structuralFormula>
        <groupCentre id="cc1" type="sp3" formula="C">
                <group id="cc1g1" type="atom" title="hydrogen" formula="H">
                        <relation title="bond">
                                <relationTarget id="cc1" title="groupCentre" />
                                <bond order="1" type="wedge" plane="above" />
                        </relation>
                </group>
        </groupCentre>
</structuralFormula>
```

Figure 10. XML code describing the relationship between a carbon center and one of its substituent group.

<relation> element associated with the <reagent> is provided with the value of "combinesWith" for its title attribute and a value of "reactant" for the title attribute in the <relationTarget> element. From this, it can be inferred that the reagent molecule is combining with the reactant molecule in the reaction. Similarly, the reversible nature of a reaction can be described by attaching the relationship between <reactantList> and <productList> elements as shown in the Figure 9.

The functionality of the <relation> element is multifaceted. This element is also used to encode the topology of substituents around a carbon center in describing the structure of a molecule. The code fragment describing the relationship between a carbon center and one of its substituent groups is shown in Figure 10. The code represents an sp^3 hybridized carbon connected to a hydrogen atom with a single bond. The topography of the single bond is described with a <bond> element associated with attributes, namely, order, type, and plane. According to the code, the bond between the carbon atom and the hydrogen atom is wedge-shaped and projecting above a reference plane.

The XML code describing a carbon center with four substituent groups is already shown in Figure 5. All four of the groups designated as 1, 2, 3, and 4 are tied with a single-bond relation to the same carbon atom as referenced by the id values. The positions of the four bonds are assumed in such a way that the groups 1, 2, 3, and 4 are occupying the top, bottom, left, and right positions, respectively, with respect to the carbon center. According to this assumption, the code describes the topography around the carbon center as the bond connecting group 4 and the carbon atom is the reference plane. The bonds connecting groups 1 and 2 with the carbon center are projecting above the plane. Group 3 is described as a hatch bond lying below the reference plane. This representation method not only describes the groups bonded to the central carbon atom in a molecule it also provides information about the bonds such as bond order, wedge/hatch/flat representation, and the plane of the bonds with respect to a reference plane of the molecule. This approach promises description of the reaction mechanisms easily and also the encoding of the stereochemical aspects of a molecule such as cis−trans and D−L forms and so forth, which are felt as difficult to encode.

AUTOMATIC REPRESENTATION AND SEMANTIC RETRIEVAL OF ALIPHATIC NUCLEOPHILIC SUBSTITUTION REACTIONS—MODEL APPLICATIONS

Chemical Ontological Support System (COSS). The first and foremost inferential operation normally performed on ontology is automatically determining whether the given concept belongs to another concept in the taxonomy. It means finding the subclass−superclass relation between two concepts. The second operation expected is the concept matching to locate a particular concept on the basis of a description. This operation enables the finding of other objects in the ontology that are structurally similar to a particular object. Another operation needed is the clustering of a set of instances to form new classes by identifying the way in which the individual instances are related. In the present work, the major operation on the ontology exploits the binary relations mainly. The XML documents describing the ontologies are clustered into a support system named the chemical ontological support system (COSS) in such a way as to derive applications out of it. As a model application, the representation of the aliphatic nucleophilic substitution reaction is attempted using COSS. The application also involves the development of a retrieval system supported with COSS capable of extracting very precisely the XML documents containing reaction components marked up with XML elements. The overall architecture of the representation and retrieval system integrating COSS is given in Figure 11.

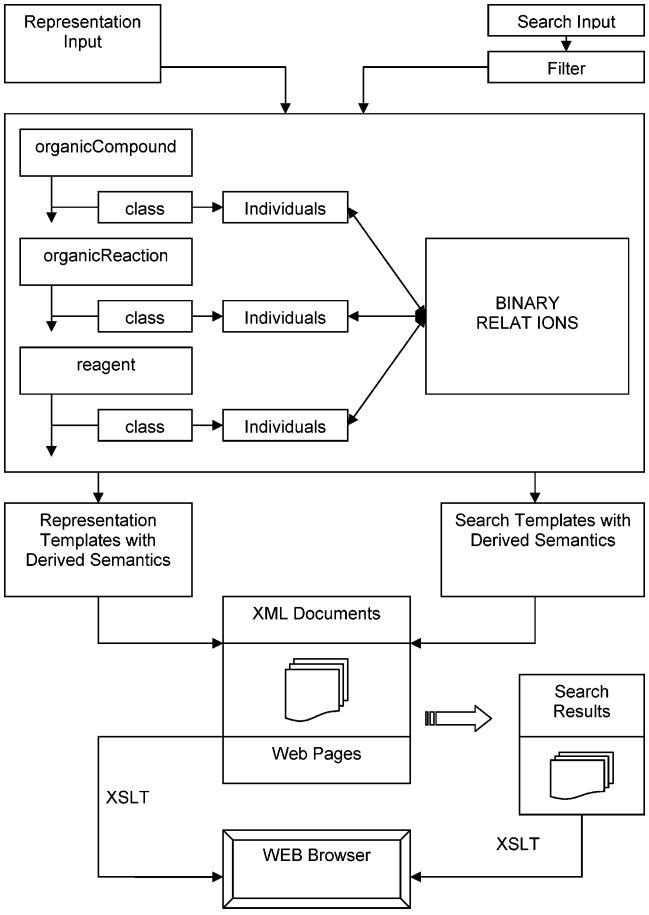

Figure 11. Architecture of automatic representation and semantic retrieval system.

Representation Model. An aliphatic nucleophilic substitution reaction can be represented as $RX + Nu^- \rightarrow RNu + X^-$, where RX is the reactant molecule, Nu^- is the nucleophilic unit acting as the substitute for the leaving group in the substrate, the main product is RNu, and the X^- is the byproduct component of the reaction. In this reaction, the substrate is an organic compound in which a portion designated as the leaving group is getting replaced by a nucleophilic unit derived from a nucleophilic reagent. Various components of this reaction are identified and

```
<reaction>
    <reactionStep id="" title="step1>
        <reactantList>
            <reactant/>
            <reagent/>
        </reactantList>
        <productList>
            <mainProduct/>
            <byProduct/>
        </productList>
    </reactionStep>
    <reactionStep id="" title="step2>
        <reactantList>
            <reactant/>
            <reagent/>
        </reactantList>
        <productList>
            <mainProduct/>
            <byProduct/>
        </productList>
    </reactionStep>
</reaction>
```

Figure 12. Skeleton structure markup for a multistep reaction.

defined suitably with XML elements with declarations in the DTD in its namespace.[30] The element <reactant> is the container for an organic molecule playing the role of a reactant in a chemical reaction. This implies that the components of a molecule such as functional group and nonfunctional group are enclosed within the reaction markup. Similarly, molecules taking the roles of reagent, main product, and byproduct are accommodated in the corresponding container elements. The inner markup of molecule resembles the ontology implementation syntax described earlier.

On the basis of the above markup style, a compact representation template is drawn for describing the aliphatic nucleophilic substitution reactions. When the template is used, a number of reactions of the above type can be represented simply by assigning the contents for the elements and providing the appropriate values for the attributes. A software agent extracts the element contents and attribute values from COSS. The retrieved information is fed into the template, and the XML document describing the specified reaction is created in the location earmarked. For reactions proceeding in a single step, the general markup structure generated by COSS includes components such as the reactant, reagent, and products. In a multistep scenario, the product formed in a single step becomes the starting material for further reactions. In such a case, the markup structure changes in such a way that each single-step reaction is marked separately under <reactionStep> elements. Thereby, the skeleton structure of the markup for a multistep reaction takes up the form shown in Figure 12. This representation format describes a reaction proceeding through more than one step by the appearance of <reactionStep> elements containing the inner markup corresponding to each step of the reaction. This results in an explicit way of showing the reaction pathways when compared to the implicit formats of SMILES and SMIRKS.

The support system COSS offers an excellent support starting from the specification of inputs for the reaction representation up to the complete markup of the specified reaction as an XML document. The markup starts with an input of specifying the reaction type to be represented. For an aliphatic nucleophilic substitution reaction, COSS allows compound classes such as aliphatic monohalogen compounds, aliphatic ethers, and so forth, which can undergo this type of substitution reaction. Subsequently, it also permits only individual compounds such as methyl chloride, ethyl bromide, and so forth as the eligible compounds, enabling the choice of a suitable reactant molecule. This makes the input data more meaningful, avoiding the unnecessary futile trials. If the input for the reactant-side specification is valid, the automatic representation process is activated. In this process, a software agent extracts the necessary values needed for the element content and attributes from COSS. Also, it derives the essential semantics required for the representation template choice. The generated values are fed into a suitable template, and the code for the XML document representing the reaction is written automatically. The resultant XML document is stored in the specified location with a unique file name. The same document can also be stored as a Web page or as a component of a Web page. The rendering of an XML document in a Web browser[38] or in a browser component is achieved using an appropriate XSLT[39] program. The representation model for an input of reaction type, reaction of alkyl halides with moist silver oxide, and ethyl bromide as the reactant is shown in Figure 13. A fragment of the code generated automatically by the system is presented in Figure 14.

Retrieval Model. The retrieval framework is based on an XML node-based search for a specified keyword filtered for its role in the reaction. The input keyword is filtered for its identity as a reactant or reagent or product and triggers the retrieval process. In this process, a keyword mapping with the contents of appropriate elements over the available XML documents is performed. For a given reactant molecule name as the search keyword, the system retrieves only those XML documents containing the reactant molecule as a reactant component of the nucleophilic substitution reaction. A search string filtered as the reagent lists all of the XML documents describing a nucleophilic substitution reaction having the specified keyword string as the reagent component of the reaction. For example, with the search keyword "aqueous potassium hydroxide" filtered as the reagent component of the reaction, COSS identifies the string "aqueous potassium hydroxide" as a reagent in the nucleophilic substitution reaction and identifies the formula with which the document is to be searched as "KOH". Holding the contents to be mapped, the system triggers an XML tag-based search over the available XML documents. During mapping, the element's name is mapped first. For every encounter of the element name, the search contents are mapped. Thus, for the encounter of the element <reagent>, the corresponding attribute values of "title" and "formula" are mapped with the search contents identified by COSS. An exact match of the contents results in the capture of the document as the search result. Accordingly, those documents in which aqueous potassium hydroxide is marked as the reagent molecule are captured as the search result. Thereby, a meaningful and precise retrieval of documents is achieved. Moreover, the documents containing similar search contents in different container elements are ignored by the search eliminating the collection of unwanted documents. The search results for the keyword "potassium hydroxide" filtered for its role as a reagent in the browser component is shown in the Figure 15.

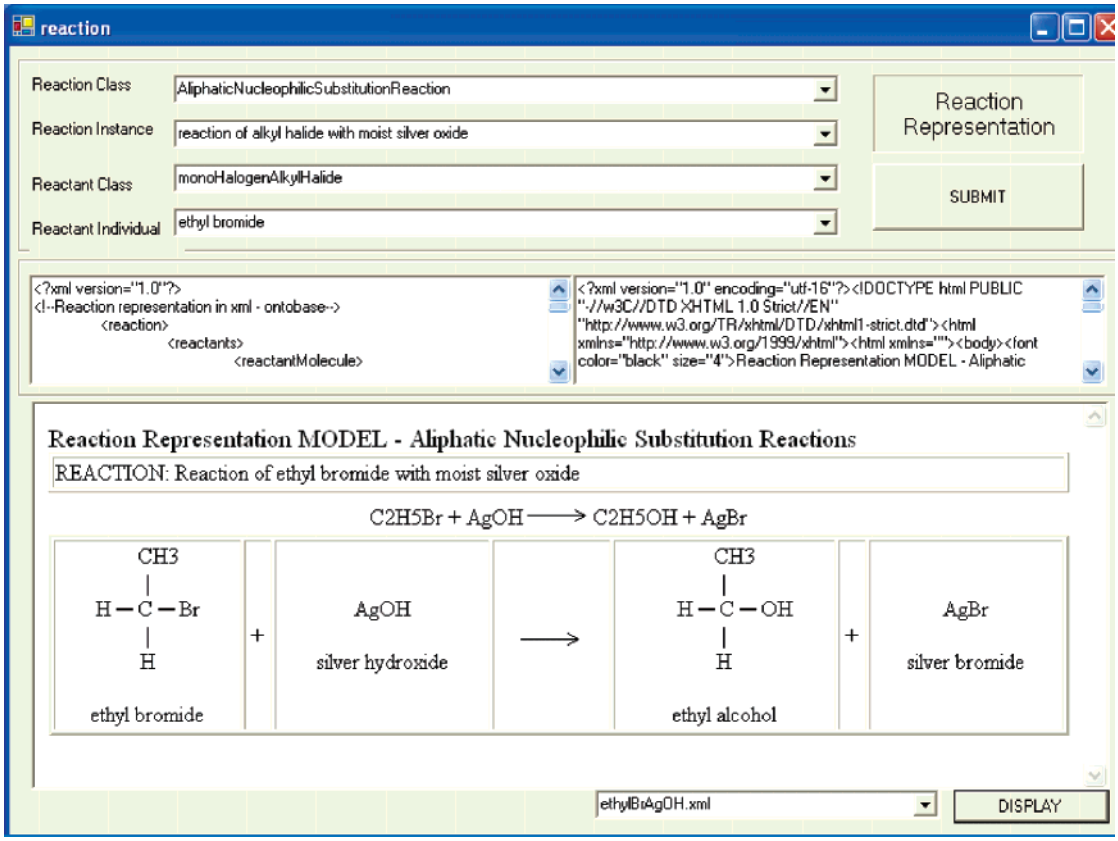

Figure 13. Reaction representation model.

```
<?xml version="1.0"?>
<?xml-stylesheet type='text/xsl' href='SNRxsl.xsl'?>
<!DOCTYPE reaction SYSTEM "rxnDTD.dtd">
<!--Reaction representation in xml - ontobase-->
<rxn:reaction xmlns:rxn = "http://www40.brinkster.com/sankarpec/rxn/"  id="R1" type="organic"
                title="aliphatic SN Reaction">
...
<rxn:productList>
   <rxn:relation title="irreversibleWith">
     <rxn:relationTarget id="" title="reactants" />
   </rxn:relation>
   <rxn:mainProduct id="m1" type="aliphatic" title="ethyl alcohol" formula="C2H5OH">
     <rxn:structuralFormula>
       <rxn:groupCentre id="cc1" type="sp3" formula="C">
         <rxn:group1 id="m1cc1g1" type="group" title="methyl" formula="CH3" />
         <rxn:group2 id="m1cc1g2" type="atom" title="hydrogen" formula="H" />
         <rxn:group3 id="m1cc1g3" type="atom" title="hydrogen" formula="H" />
         <rxn:group4 id="m1cc1g4" type="fng" title="hydroxide" formula="OH" />
       </rxn:groupCentre>
       <rxn:functionalGroup id="hydroxide" type="alcohol" title="hydroxide" formula="OH" />
       <rxn:non-functionalGroup id="ethyl" type="pri" title="ethyl" formula="C2H5" />
     </rxn:structuralFormula>
   </rxn:mainProduct>
   <rxn:byProduct id="" type="salt" title="silver bromide" formula="AgBr" count="1">
     <rxn:relation title="formsWith">
       <rxn:relationTarget id="m1" title="mainproductMolecule" />
     </rxn:relation>
   </rxn:byProduct>
 </rxn:productList>
```

Figure 14. Fragment of the XML document generated automatically by COSS.

In the context of the Semantic Web, the storage, retrieval, and communication of chemical structures are to be handled by markup languages. The molecular structures represented in markup language are a convenient means to represent chemical reactions because the representation of reactions in turn is based on the structure description. Because the search and harvest of chemical reactions can be performed effectively by incorporating the needed semantics for reaction components in terms of their role and relationships, the structure description is made up to this level. The markup procedure itself provides the semantics for the role of reaction components such as reactant molecule, reagent molecule, main product, byproduct, and so forth. The further addition of semantics for inner details of the structure with elements describing the functional group, nonfunctional group, group, and so forth present in the structure brings more advantage

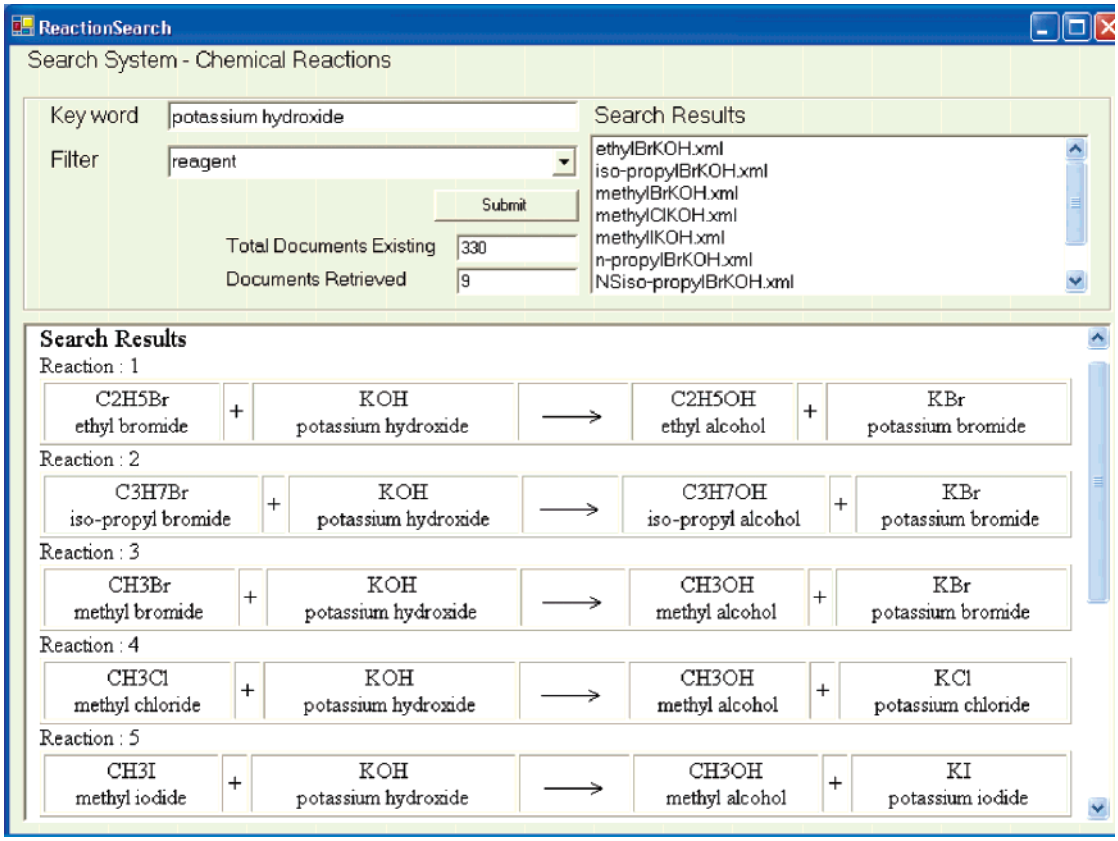

Figure 15. Search result.

for the search process. As such, the system is capable of harvesting the documents describing chemical reactions by searching for the reaction components filtered for their role in the reaction. However, this can be extended to search the reaction components with specific inner structural features. This approach works excellently not only in retrieving precise documents but also in eliminating irrelevant documents, making the retrieval process meaningful.

Both the representation and retrieval models proposed are fully based on XML technology in which no conventional database and associated SQL activities are involved. At present, the models seem to be simple applications as evidenced by the screen captures (Figures 13 and 15). However, the transformation of these applications into Web-based systems is quite possible. Because the reactions are represented as XML documents, they can be viewed in a Web browser by embedding the XML file in an HTML program. Also, the system allows the flexibility to change the display of the represented reactions by changing the associated XSLT program. As a stand-alone system, a Web developer can generate Web pages containing reaction components using the above method. The system in its matured stage, in the future, will allow the COSS to be available on a Web server so that, by knowing the syntax, applications can be drawn through the Internet. The conversion of the proposed retrieval model into a Web-based system is not so straightforward because of some preconfigured activities involved in search and retrieval processes. In a search process, the query is submitted to the search engine; then, a Web crawler crawls the indexed pages with tag names and indexes supplied by the search engine. Matching documents are then retrieved and displayed on the CRT by the browser. At present, the proposed retrieving system suffers from these preconfigured activities associated with the search process in becoming a Web-based system. However, this difficulty can be eliminated by providing suitable plug-in components.

CONCLUSION

A reaction representation fully supported by chemical ontologies is very comfortable when compared to the database-oriented methods. The ontology implementation in a Web-based meta language like XML provides a platform that can be efficiently and seamlessly integrated with multiple applications in the chemical domain. Though the boundaries of ontologies are confined to support the model applications, they can be expanded with suitable definitions. Also, the COSS for the model application promises the development of many software agents for reuse of the knowledge repository. The automatic representation and retrieval systems, when suitably modified, can be effectively used in generating Web pages containing reaction components. Also, it ensures the development of Web-based teaching tools describing the chemical reactions. The approach of chemical markup using relations paves the way for encoding chemical contents which are thought to be difficult.

ACKNOWLEDGMENT

This research work is supported by the Research Grant from All India Council for Technical Education (AICTE), New Delhi, India, under Research Promotion Scheme (RPS) 2004-2006 F.No. 8022/RID/NPROJ/RPS-109/2003-04.

Supporting Information Available: A PDF file consisting of extensive classification of the reactions, compounds, and

regents. Sample source code for the formalization of the reaction class, compound class, reagent class, and individual instances in each case are included. The file also contains the complete DTDs and an XML document generated automatically representing the aliphatic nucleophilic substitution reaction. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Studer, R.; Benjamins, V. R.; Fensel, D. Knowledge Engineering: Principles and Methods. *Data Knowl. Eng.* **1998**, *25* (1−2), 161−197.

(2) Chandrasekaran, B.; Josephson, J. R.; Benjamins, V. R. What Are Ontologies, and Why Do We Need Them? *IEEE Intell. Syst.* **1999**, *14* (1), 20−26.

(3) Murray-Rust, P.; Rzepa, H. S. Markup Languages How to Structure Chemistry-Related Documents. *Chem. Int.* **2002**, *24* (4), 9−13.

(4) Feldman, H. J.; Dumontier, M.; Ling, S.; Haider, N.; Hogue, C. W. V. CO: A Chemical Ontology for Identification of Functional Groups and Semantic Comparison of Small Molecules. *FEBS Lett.* **2005**, *579*, 4685−4691.

(5) Fernández-López, M.; Gómez-Pérez, A.; Pazos-Sierra, J.; Pazos-Sierra, A. Building a Chemical Ontology Using Methontology and the Ontology Development Environment. *IEEE Intell. Syst.* **1999**, *14* (1), 37−46.

(6) Stanford KSL Network Services Home Page. http://www-ksl-svc.stanford.edu:5915 (accessed Mar 30, 2005).

(7) Ontolingua. http://ksl.stanford.edu/software/ontolingua (accessed Mar 30, 2005).

(8) Loom Project Home Page. http://www.isi.edu/isd/LOOM/LOOM-HOME.html/ (accessed Apr 15, 2005).

(9) OCML: Operational Conceptual Modeling Language. http://kmi.open.ac.uk/projects/ocml/ (accessed Apr 15, 2005).

(10) SHOE: Simple HTML Ontology Extensions. http://www.cs.umd.edu/projects/plus/SHOE/ (accessed Apr 15, 2005).

(11) RDF Vocabulary Description Language, Version 1.0: RDF Schema W3C Recommendations 10 February, 2004. http://www.w3.org/TR/rdf-schema/ (accessed Apr 15, 2005).

(12) XOL Ontology Exchange Language. http://www.ai.sri.com/pkarp/xol/ (accessed Apr 15, 2005).

(13) Ontology Markup Language Version 0.3. http://www.ontologos.org/OML/OML%200.3.htm (accessed Apr 15, 2005).

(14) Description of OIL. http://www.ontoknowledge.org/oil/ (accessed Apr 15, 2005).

(15) DAML Language. http://www.daml.org/language/ (accessed Apr 15, 2005).

(16) McGuinness, D. L.; Fikes, R.; Hendler, J.; Stein, L. A. DAML+OIL: An Ontology Language for the Semantic Web. *IEEE Intell. Syst.* **2002**, *17* (5), 72−80.

(17) Murray-Rust, P.; Rzepa, H. S. Chemical Markup Language and XML Part I Basic Principles. *J. Chem. Inf. Comput. Sci*. **1999**, *39*, 928−942.

(18) Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML and the World-Wide Web. 2 Information Objects and the CML DOM. *J. Chem. Inf. Comput. Sci*. **2001**, *41*, 1113−1123.

(19) Gkoutos, G. V.; Murray-Rust, P.; Rzepa, H. S.; Wright, M. Chemical Markup, XML and the World-Wide Web, Part III: Towards a Signed Semantic Chemical Web of Trust. *J. Chem. Inf. Comput. Sci*. **2001**, *41*, 1124−1130.

(20) Murray-Rust, P.; Rzepa, H. S.; Wright, M. Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content. *New J. Chem.* **2001**, *25*, 618−634.

(21) Holliday, G. L.; Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML, and the World Wide Web. 6. CMLReact, an XML Vocabulary for Chemical Reactions. *J. Chem. Inf. Model.* **2006**, *46*, 145−157.

(22) Gorden, J. E. Chemical Inference. 3. Formalization of the Language of Relational Chemistry: Ontology and Algebra. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 100−115.

(23) CambridgeSoft chemDraw. http://www.cambridgesoft.com/ (accessed Feb 24, 2006).

(24) MDL ISIS/Draw. http://www.mdl.com/ (accessed Feb 24, 2006).

(25) SMILES—A Simplified Chemical Language. http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html (accessed Feb 24, 2006).

(26) SMIRKS Tutorial. http://www.daylight.com/dayhtml_tutorials/languages/smirks/index.html (accessed Feb 24, 2006).

(27) REACTOR. http://www.chemaxon.com/chem/doc/user/Reactor.html (accessed May 7, 2006).

(28) W3C—Scalable Vector Graphics (SVG). http://www.w3.org/Graphics/SVG/ (accessed May 10, 2006).

(29) W3C—Math Home. http://www.w3.org/Math/ (accessed May 10, 2006).

(30) GAPSPEC. http://www40.brinkster.com/sankarpec/onto/ and http://www40.brinkster.com/sankarpec/rxn/ (created May 7, 2006), namespaces.

(31) OntoWeb. http://www.ontoweb.org (accessed Feb 24, 2006).

(32) IEEE *xplore*—IEEE Standard for Developing Software Life Cycle Processes. http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=10452 (accessed May 7, 2006).

(33) The Foundation for Intelligent Physical Agents. http://www.fipa.org/ (accessed Feb 24, 2006).

(34) Finar, I. L. Halogen Derivatives of Alkanes. In *Organic Chemistry*, 6th ed.; PEARSON Education: Singapore, 2003; Vol. 1, pp 145−175.

(35) Morrison, R. T.; Boyd, R. N. Alkyl Halides Nucleophilic Aliphatic Substitution. In *Organic Chemistry*, 6th ed.; Prentice Hall of India Private Limited: New Delhi, India, 1998; pp 165−212.

(36) Bruckner, R. Nucleophilic Substitution Reactions at the Saturated C Atom. In *Advanced Organic Chemistry—Reaction Mechanisms*; Academic Press, An Imprint of Elsevier: New Delhi, India, 2005; pp 43−83.

(37) IUPAC Compendium of Chemical Terminology—Online "Gold Book". http://www.chemsoc.org/chembytes/goldbook/index.htm (accessed May 7, 2006).

(38) Microsoft Internet Explorer Version 6.0. http://www.microsoft.com/Windows/ie/ie6/default.mspx (accessed May 7, 2006).

(39) XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November, 1999. http://www.w3.org/TR/1999/REC-xslt-19991116 (accessed Feb 24, 2006).

CI050533X