An advanced learning and discovering system

J.B.Savkovic-Stevanovic

Abstract: - This paper presents a new approach and method for learning and discovering. Concept learning illustrates acquiring descriptions that make the structure of generalizations explicit. Two kinds of concept learning are distinguished inductive and deductive. The entity structure in which all components are encoded has modeled knowledge representation scheme. Discovering with process knowledge acquisition and learning were applied. Learning by discovering systems enhance their knowledge base. In this paper examines the entity structure for chemical system building in organic chemistry. The obtained results has illustrated the development of discovering model for surfactants formation.

Key-Words: - Concept learning, process discovering, compounds, functional groups, formulation.

I. ITRODUCTION

Learning by discovering is when systems operate autonomously, performing experiments to enhance their knowledge base. They can proceed either deductively or inductively. They search for interesting things, forming conjectures by examining a few examples of concepts and noticing coincidences, perhaps attempting to prove these conjectures[1]-[4]. One key issue is the mechanism by which interesting conjectures are singled out and others discarded. Another is the distribution of resources between competing lines of development. It is not yet clear whether such open ended search is capable of being controlled usefully.

Concept learning is rendered tractable by constraining the search to exclude major portions of the potential space. In existing systems the designer does this by ruling out possible forms of a concept. Ideally, a mechanism would be provided whereby a knowledge acquisition system could learn about the constraints which operate in a particular domain, but the general problem of learning constraints has yet to be tackled. Note that amenable to a purely formal treatment, for biasing constraints can only be expressed in logic as properties of predicates, so reasoning about bias entails the use of higher order logic.

The scientists view toward causality differs considerably from that of philosophers. Scientists are interested in discovering functional relationships among physical phenomena in order to explain their behavior. Over the years, scientists have studied two aspects of causality: isolation of the variables which represent cause phenomena and those which represent effect phenomena, and determination of the magnitude and direction of change in effect phenomena corresponding to a change in cause phenomena [5]-[8].

II. CONCEPT LEARNING

Inductive and deductive concept learning can be represented in an appropriately powerful form of logic, such as predicate calculus. However, although formal logic provides a sufficient basis for deduction, as a foundation for induction it is at once too narrow and too powerful.

J.B.Savkovic-Stevanovic,Department of Chemical Engineering, Faculty of Technology and Metallurgy,University of Belgrade, Karnegijeva 4, 11000 Belgrade,Serbia, e-mail:savkovic@tmf.bg.ac.rs http: //www. tmf. bg.ac.rs

It is too narrow because induction, based on search, is extraordinarily sensitive to the precise structure of the search space, and while different representations may be formally equivalent, they imply different search spaces.

Extending the distinction that is often made between the epistemiological adequacy of a representation whether it is capable of expressing the facts that one has about the world and its heuristic adequacy, one might characterize formal logic as epistemiologically adequate but inductively inadequate for concept learning

On the other hand, logic is too powerful because the need to acquire knowledge automatically from teacher or environment and integrate it with what is already knows means that only the simplest representations are used by programs for system learning. It is hardly necessary to say that a data structure is not knowledge, any more than an encyclopedia is.

Current knowledge acquisition systems perform routine housekeeping, permit rote learning of explicitly presented facts, and are able to elicit from experts simple rules based on the attributes. Methods of concept learning may be able to overcome these imitations, although the present state of the art is primitive and suggests ideas rather than well developed algorithms for the knowledge acquisition tool box. Concept learning systems take examples and create general descriptions, often expressed as rules, which expert systems need.

Dictionary definitions of concept are remarkably vague, but have in common the abstract idea of a class of objects, particularly one derived from specific instances or occurrences. Learning is an equally broad term, and denotes the gaining of knowledge, skill, and understanding from instruction, experience, or reflection in other words, knowledge acquisition by people.

Others have defined terms such as generalization inductive learning and inductive modeling in almost identical ways. Moreover, what is learned is sometimes called a generalization, a description, a concept, a model, or a hypothesis. A satisfactory technical vocabulary has not yet been developed, which term one favor seems purely a matter of taste.

Concept learning involves acquiring descriptions that make the structure of generalizations explicit. When a person learns a new concept, he or she gains knowledge that they can use in a rich variety of different ways, apply it to other ideas, and so on. For a person, learning is not simply a matter of acquiring a description, but involves taking something new and integrating it fully with existing though processes. However, they do acquire descriptions that are explicit in the sense that they can be communicated economically for example, in the form of rules, and could plausibly support a variety of different kinds of reasoning. This orientation rules out, for example, connectionist model of learning, which embed knowledge in high dimensional numerically parameterized spaces, making learning a process of weight adjustments.

While the phrase may conjure up an alluring promise of the magic of human intelligence, systems for concept learning are rooted firmly in the reality of practical algorithms. The essence is a constrained search of what is invariably an astronomically large space of possible descriptions. The framework can classify into areas of representing concepts and examples, biasing the search for concepts and interacting with a teacher.

There are two kinds of concept learning inductive and deductive. For the inductive concept learning, one must infer a general conclusion from empirical examples [4]. This can be formalized as follows. Along with initial background knowledge -BK, examples -E of a desired concept are given. A concept C is said to be induced if:

$BK \neq > E$ the examples are not logical consequences of the knowledge

$BK \cup E \neq > -C$ the concept is not inconsistent with the knowledge and examples

$BK \cup C \equiv > E$ the examples are logical consequences of the concept and knowledge together

Normally $BK \cup E \neq > C$, C is induced rather than deduced. If E is known to be exhaustive, the learner may be able to deduce a concept. The inference of a statement from information that is known to exhaust all possibilities is a special case of induction known as summative induction, and lies on the borderline with deduction[4],[5].

Suppose examples and concept are both deducible from background knowledge, even though neither are explicitly, present in it. This is an instance of deductive concept learning (Fig. 1). Although, excluded by the above formalization, it can be viewed as a kind of learning, for the examples show which deductions are important namely those that represent the reasoning involved in the examples and enable this important knowledge to operation in an explicit description. In most cases an impossibly large set of operational concepts can be deduced from background knowledge, and the problem is to select one appropriate to the examples.

Deductive concept learning may be formalized as follows:

$BK => E$ the examples are logical consequence of the background knowledge

$C \notin BK$ the concept is not an explicit part of the background knowledge

$BK => C$ the concept is the logical consequence of the background knowledge

$C => E$ the examples are the logical consequence of the concept.

In other words, the concept operationalizes the relevant part of background knowledge. This kind of learning is often called explanation based. The distinction between deductive and inductive concept learning can be viewed as a modern reincarnation of the long philosophical tradition of distinguishing necessary from contingent truths. Concepts and examples are the output and input of the knowledge acquisition system, what is learned and what is provided by a teacher or other external agent (Fig.2). To be useful, a framework for representing concepts must provide knowledge engineers with methods for selecting appropriate representations for examples, concepts and background knowledge. Separate representations are required for examples and concepts.

Methods for determining a preference order inevitably depend on the syntax of the language used to express concepts. One concept is simpler than another if it is shorter when written down. Length can be measured in various ways including number of disjuncts in logical expression, number of function calls in functional expressions, number of function calls plus total number of function arguments, length of function expressions represented as character strings, number of loops and conditionals in procedures, and number of states or transitions in a state diagram representation.

A different approach model maximization uses the extension rather than the intension of a concept. One concept is preferred over another if its extension is a superset of the others. The knowledge acquisition system must know how to order concepts according to their extensions, the ordering will be partial as extensions may overlap (Fig.3).

A concept learning system practical utility obviously depends critically on its teaching requirements. Teaching differs from programming in that a teacher does not use a formal model of the learner, whereas a programmer normally expects to know exactly how his instructions are going to be interpreted. In other words, a teacher adopt the intentional stance while a programmer adopts the design stance. Ideally, therefore, to be a good teacher one need know nothing about the internal structure of the learner, no about the representation it uses for concepts.

Moreover, experts who acquire their knowledge from example cases are unlikely to have a clear analytical understanding of the task domain, otherwise more explicit methods of knowledge transfer will probably prove more appropriate. In general, people find it difficult to translate their own expertise into explicit descriptions. Consequently concept learning systems should be able to work with the kind of examples that a teacher finds it natural to provide, ordered in way that is natural to him.

The third concerns the cooperation and pedagogical skills of the teacher. Learning situations range from having no teacher at all through a naive user, a domain expert, a skilled trained teacher to a teacher who is prepared to program in a conventional programming language and thereby circumvent the need for learning. Must the teacher show all working give examples, teach simple concepts before complex ones.

A skilled teacher will select illuminating examples himself and thereby simplify the learners task. The benefits of carefully constructed examples were appreciated in the earliest research efforts in concept learning. The notion of sympathetic teacher has been formalized in terms of felicity conditions, constraints imposed on or satisfied by a teacher that make learning better than from random examples. The teacher should classify examples correctly, point out absences explicitly, show all work, avoid glossing over intermediate results, and introduce only essentially new feature per lesson. Although it does not increase formal learning power, the possibility of a system constructing its own examples and having them classified by an informant has a system considerable potential to speed up learning and reduce dependence on the skill of the teacher.

Isolated attempts have been to meet what is perhaps the major shortcoming of similarity based learning, that examples are lifted out of the world, cleaned up, and presented to system by a teacher who makes all the important decisions about when to create new concepts. Still the most impressive to discover interesting concepts in elementary mathematics and interesting heuristics for further discovery.

What is certain to be required is an integrative framework that permits different mechanisms to be compared and contrasted learning techniques in a way that makes them readily accessible to potential users. Anything that is learned must be communicated through the expert or user interface.

Faced with practical problem, the first decisions to make are how to represent concepts and examples. Suitable forms of concepts representation will be dictated by the requirements of the knowledge based system and the kind of examples available. Sometimes the example representation dominate the decision, while in other situations the desired concept format will force examples into a particular mode.

Logical representations are indicated by the predominance of logical relationships in example or concepts. The possibility of using attribute values strongly suggests the simpler prepositional calculus representation. Similarity based system methods are appropriate when many examples are available, or when it is not possible to define a domain theory in advance. If concept must be built on earlier learned ones, a hierarchical method is indicated. If a domain theory is known, then an explanation or discovery based system can utilize it.

Given a set of objects that represent examples and counterexamples of concept, a similarity based learner attempts to induce generalized description that encompasses all the examples and none of the counterexamples. Interesting general issues include the conditions under which the procedure converges to a single description, weather the system can know that it has converges, whether the final concept may depend on the order of presentation of examples, and whether the training sets expected to be exhaustive or representative.

The version space approach to concept learning transforms the inductive problem of generalization into a deductive one by circumscribing the way in which descriptions are expressed and searching for ones that fit the examples given. It postulates a language in which objects are expressed. Given asset of positive and negative examples of a target description, a simple search algorithm exists that finds all descriptions that are consistent with the examples (Eq.1). This set is called the version space.

$$E=f(concept1,concept2\dots, example1,example2,\dots\dots bias) \quad (1)$$

The seeking value E of introduced concept and bias and examples is a major task. Fig.1 shows trajectory for concept introducing in discovering process. The method applies simply and directly when each object is described by a fixed set of properties, usually epresented as an attribute vector, which is equivalent to a description in propositional logic. Its performance in such domains has been studied extensively. Allowing disjunction in the description language causes the version space to explode, while even with purely conjunctive concepts, one version space boundary can grow exponentially in the number of examples.

In structural domains, each example comprises a scene containing several objects, expressed in predicate logic. Part of the problem in matching a scene with a structural description is determining an appropriate mapping between objects in the scene and those specified in the description. This mapping will have different interpretations depending on whether the scene is to compromise or merely to contain the desired object. Several theoretical results indicate that, even in the simplest cases, extreme computational complexity can be involved in working with version spaces of structural objects.

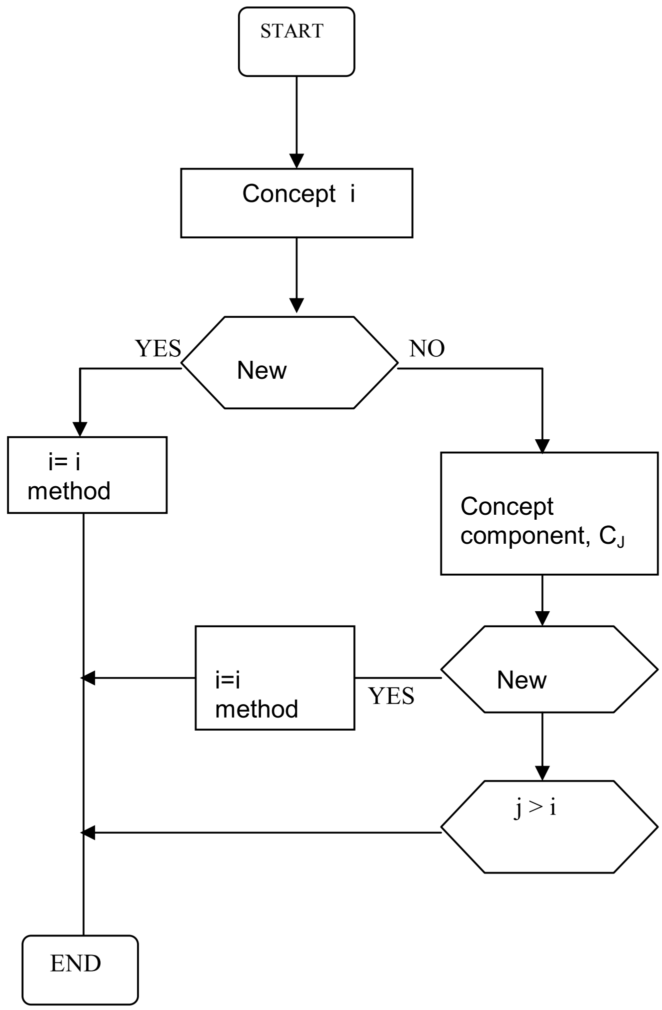

Fig.1 Concept development

Fig.3 shows model for knowledge extension.

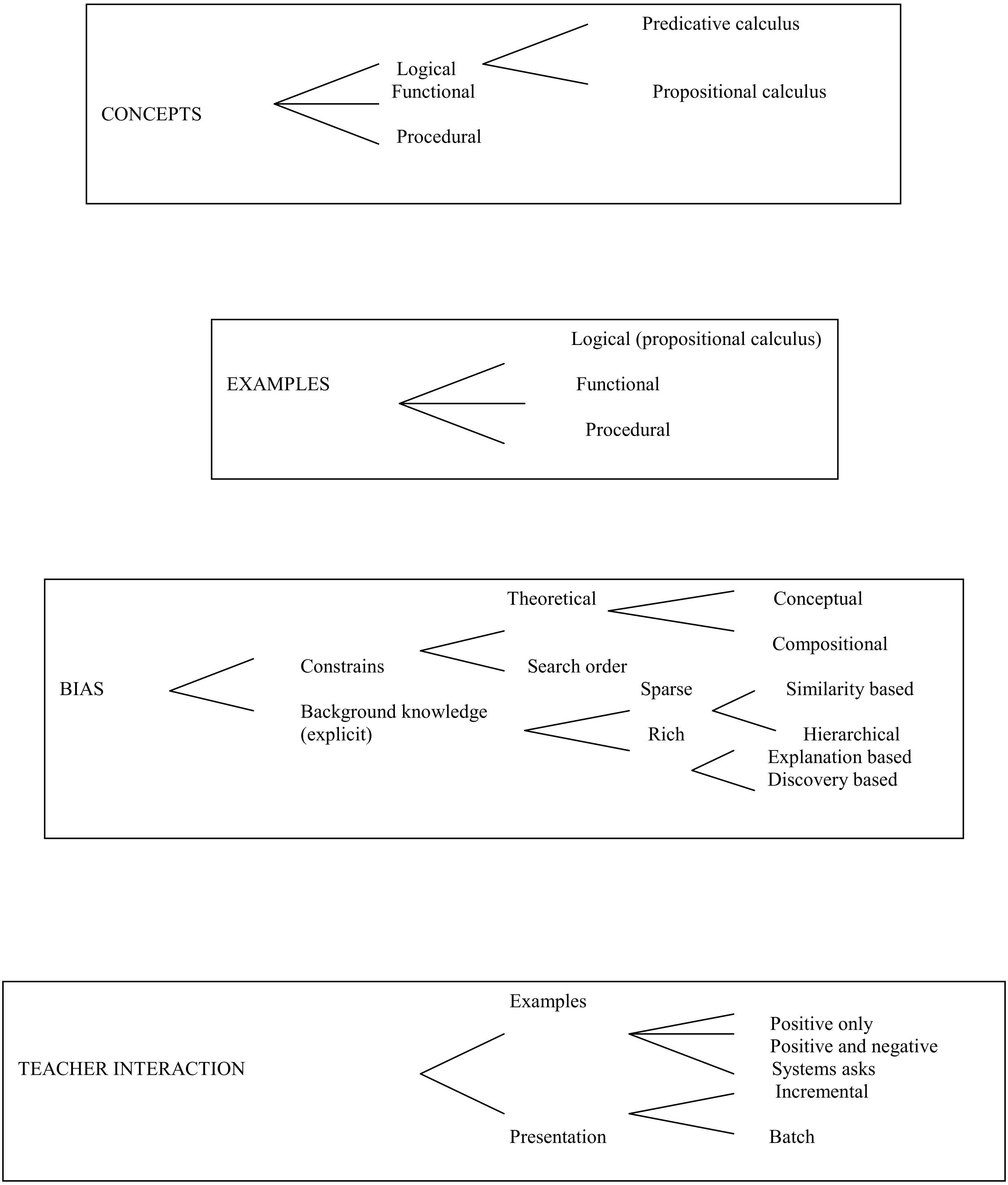

Fig.2 Learning method

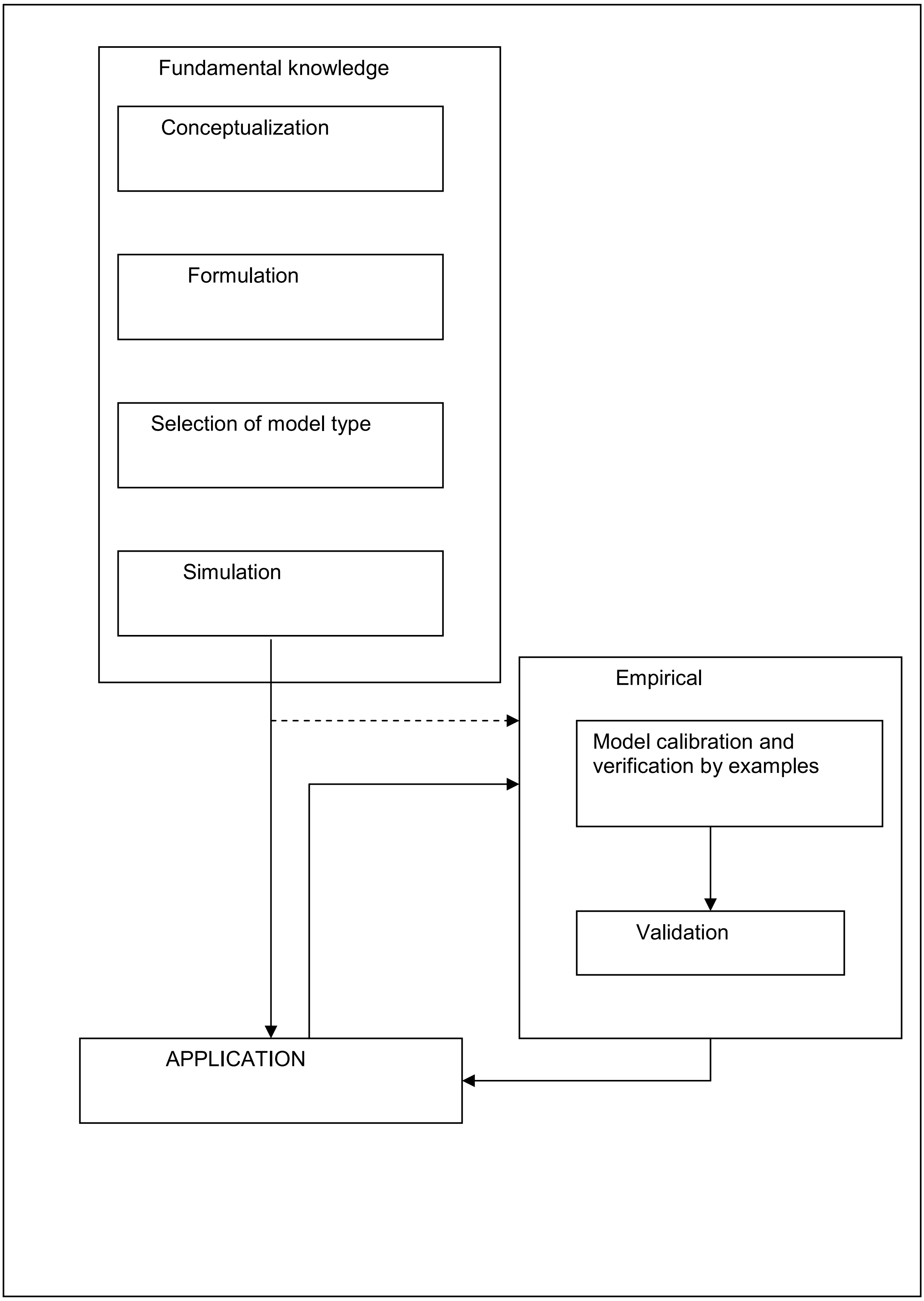

Fig.3 Model construction

III. CURRENT AND PREVIUOUSLY PHENOMENA

It is not possible, in the state present of the art, to supply a comprehensive set rules to determine which concept learning scheme to use, given desire concept and example representations, background knowledge available, form of teacher interaction, and appropriate biases for concept representations. The scientists view toward causality differs considerably from that of philosophers. Scientists are interested in discovering functional relationships among physical phenomena in order to explain their behavior. Over the years, scientists have studied two aspects of causality: isolation of the variables which represent cause phenomena and those which represent effect phenomena, and determination of the magnitude and direction of change in effect phenomena corresponding to a change in cause phenomena [7]. The identified variables along with their functional relationships can serve as a useful computational model for making inference. Causal structure can be establish by examining some statistics associated with the variables of interest. The question "Is correlation proof of causation?" is very important. Causal modelling attempts to resolve questions about possible causes so as to provide explanations of phenomena as a results of previous phenomena . The quantitative approach to causal modelling and inference involves computing the path coefficients between the cause and the effect variables and using the resulting equation to predict the change in the cause. It is a technique for selecting variables which are potential determinants of effects and then attempting to isolate the separate contributions to the effects by each cause [8]. The two areas of model development and analysis are addressed through the discussion of generic simulation environment. The knowledge based simulation environment is an expression of some control law or cognitive theory. To the extent that the rule base is derived from set of assumptions about the environment and performance expectations, it is a belief system. However, in the existing form, the goals are not expressed and the underlying assumptions are not evident. Consequently, they are opaque to the analyst and cannot be directly applied to the learning process. When expressed in hierarchical form the relationship that exist between goals and subgoals provide a basis for relating overall goal based system performance to specific assumptions about the variability and contribution of the supporting subgoals [9]. In this form, the belief system is a full expression of some control theory in that the system's relationship with the environment, as expressed in a set of feasible state conditions, can be related either in overall system performance measures to be relationships and the subgoals that support them. In the case of a tree, the root specifies an attributte to be selected and tested first, then depending on its value subordinate nodes dictate tests on further attributes. The leaves are marked to show the classifications of the objects they represent. For two class problems these are simply »positive«or »negative«, but it easy to distinguish more than two classes. The Quinlan's algorithm [3] uses an information theoretic heuristic to find a simple tree which classifies all examples given. With noisy data, it construct huge decision trees which reflect the detail of every example seen. In the case of production rules, the training set is used to construct a set of rules which can be interpreted by an expert system in standard forward or backward chaining manner. While any decision tree can easily be converted to rules, the rules may contain redudances which can be eliminated by generating them directly from the examples. It would be attractive if learning systems could build upon already learned and use them as components in newly constructed descriptions. This might allow learning to be sustained over an extended period of time, insteaded of being done on a one off basis.

IV. KNOWLEDGE EXTRACTION

The quantitative approach to causal modeling and inference involves computing the path coefficients between the cause and the effect variables and using the resulting equation to predict the change in the cause (Fig.4). Historically, this is the most important approach for analyzing causality and has already been explored to great extent in sociology, economics, medicine and etc..

Although the quantitative approach has proven very useful for dealing with many real-world problems, it is neither sufficient nor necessary under some circumstances. Furthermore, because in reality, there may not exist enough quantitative knowledge to permit full quantitative modelling, abstract qualitative models are worthwhile to explore. Qualitative causal modeling has become one major line of research toward the representation of deep models in knowledge based systems. Two well qualitative causal simulation techniques will be describe here.

The first approach refers to the technique which predicts the possible qualitative behaviors of a system on the basis of the model comprising the predefined physical parameters and constraint predicates.

These parameters and constraints are abstracted from the mathematical equations describing the system dynamic behavior. This model provides a snapshot of the qualitative characteristics of the system at each defined time frame, and is especially useful when we want to know the dynamic trends of the cause and the effect.

In the second approach causal knowledge is modeled as causal networks in which the nodes represent propositions (or variables), the arcs signify direct dependencies between the linked propositions, and the strengths of these dependences are quantified by conditional probabilities. Stochastic simulation is a method of computing probabilities by counting the fraction of time an event occurs in a series of simulation run. If a causal model is available, it can be used to generate random samples of hypothetical scenarios that are likely to develop in the domain. The probability of an event or any combination of events can be computed by recording the fraction of time it registers true in the samples generated.

This approach can arrive at stationary estimates of a posteriori probabilities based on a priori probabilities. As a contrast, in quantitative approach, the probability factors are already incorporated in regression coefficients.

V. DISCOVERING SYSTEM MODELLING

In this article multifaceted modelling, knowledge based simulation and causality were demonstrated. This work was motivated by the need to provide a more flexible than existing approaches modelling framework for simulating changes and for predicting the behaviour of the system in general.

Causal modelling attempts to resolve questions about possible causes so as to provide explanations of phenomena as a results of previous phenomena. It is a technique for selecting variables which are potential determinants of effects and then attempting to isolate the separate contributions to the effects by each cause. Causal ordering is an important part of causal modelling. It is an asymmetrical relationship among variables of a set of simultaneous equations representing functional relationships. The objective of causal ordering is to establish direct causality or determine the number of intervening levels for indirect causality.

Many causal models have been developed without losing generality, a causal model consists of a representation of the phenomena along with directions indicating the cause - and -effect relationship among the phenomena. A direct graph is a means of how this causal structure can be visualized, in which a node is labeled by domain concept and an arc points from a cause node to an effect node. Since to perform causal reasoning requires the knowledge of the quantitative and qualitative characteristics of causal relationships and the interaction manner among causal influences.

Each arc associates with function describing the characteristics of the designated causal relationship and associate each node with a function for combining causal influences from different sources. A formal description of this visual representation for causal structures is given in Fig.4.

Each variable X_i assumes either symbolic or numerical values. As shown in Fig.4, each arc in the network is labeled (associated) with an influence function. If function f_{ij} is associated with arc X_i X_j, the influence of variable X_i on variable X_j is $f_{ij}(X_i)$. Each such influence function f_{ij} is specified by the qualitative and, or quantitative relationships between $X_i \longrightarrow$ and X_j,. Each variable is associated with a fusion function. Suppose fusion function g_j is associated with variable X_i, which receives influences from variables X_1 , X_2 , X_3 ,.... X_k with corresponding influence function f_{1j} , ,f_{2j} , f_{3j} ,........f_{kj} respectively, then

$$X_i = g_j(f_{1j}(X_1), f_{2j}(X_2)....f_{kj}(X_k)). \quad (1)$$

Composing all the influence functions:

$$X_j = G_j(X_1, X_2, ..., X_k) \quad (2)$$

In a dynamic system, let further consider the time dimension and modify the above expression into

$$X_j(t) = G_j(X_1(t), X_2(t), ..., X_k(t)) \quad (3)$$

If there is a differentiation operator on the left hand side, it can be eliminated by taking integration on both sides. If there is a circular loop, it be removed by transforming the causal network into a form without circularity. The notion of causal lag can also be integrated into the above form. Function G_j can represent any quantitative or qualitative mapping. Thus, equation (3) may represent a single or set of equations or rules.

Causal knowledge in the given can be represented at multiple levels of abstraction. The traversal mechanism between levels must be defined, which should provide correspondence in knowledge between levels and should make discovery. The model representational methods can also be organized as a taxonomy. Process abstraction allows simulationists to construct models composed of a set of interconnected levels. Each level in the network represents the process at some given level of abstraction and is encoded using a model type appropriate to that level.

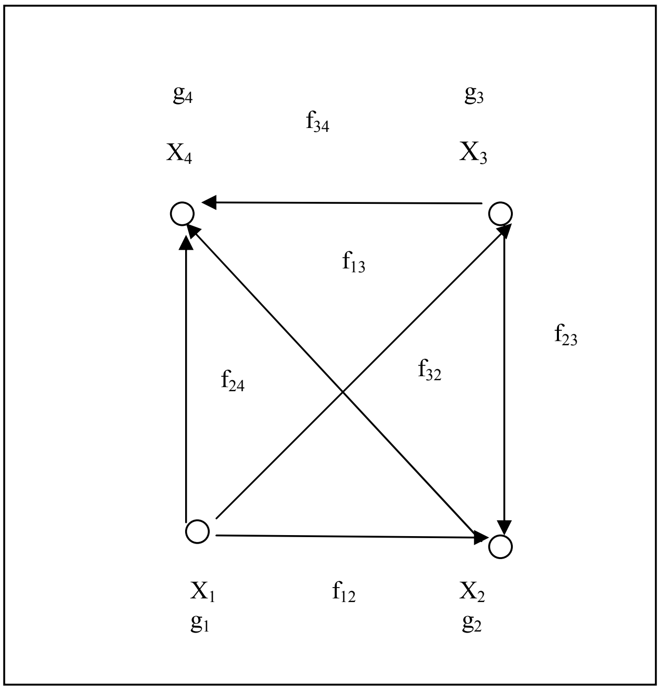

Fig.4 A graph with functional labels.

A taxonomy of process abstraction methods is introduced to characterize the fundamental concept of level traversal. Traversal mechanisms can be implemented as rules which discovery making.

In causal modeling, the quantitative method and the qualitative method treat as two different levels of abstraction. Translation quantitative knowledge into qualitative knowledge need to define. Traversal mechanisms need to discover new rules.

VI. APPLICATION IN CHEMISTRY

The framework used in modelling of one type of natural systems, structural chemical system can be depicted. It can be indicated two possible approaches to complex chemical contains modelling. The first, identified with structural knowledge, follows a deductive reasoning approach in which one tries to deduce from an existing theory model relationships for a given problem. The second, identified with a posterior empirical knowledge, follows an inductive approach in which one tries to develop a model from the sampled data. Ideally, these two approaches act as complementary stages of the modelling process. One follows all six steps with model calibration and model validation serving as an empirical test bed for a prior model as a learning tool. Yet, in some situations characterized by difficulties in obtaining empirical data due to a budget and time constraints or peliminary scope of the analysis, the model specification may be reduced to the a priori stage.

In this paper, it was demonstrated how modelling and knowledge based simulation can integrate steps required to model chemical structure of organic compounds. This work motivated by the need to provide a more flexible than existing approaches modelling framework for simulating changes in chemical structure in particular, and for predicting the behavior of chemical systems. For the application domain, chemical system simulation has been selected carbon, hydrogen and oxigen atoms generaly.

VII. LEARNING BY DISCOVERING

Learning by discovering is when systems operate autonomously, performing experiments to enhance their knowledge base. They can proceed either deductively or inductively. They search for interesting things, forming conjectures by examining a few examples of concepts and noticing coincidences, perhaps attempting to prove these conjectures. One key issue is the mechanism by which interesting conjectures are singled out and others discarded. Another is the distribution of resources between competing lines of development. It is not yet clear whether such open ended search is capable of being controlled usefully.

One key issue is the mechanism by which interesting conjectures are singled out and others discarded. Another is the distribution of resources between competing lines of development. It is not yet clear whether such open ended search is capable of being controlled usefully.

Note that amenable to a purely formal treatment, for biasing constraints can only be expressed in logic as properties of predicates, so reasoning about bias entails the use of higher order logic.

In functional representation composition bias determines the vocabulary of built in functions and operators, and the syntax by which concepts may be constructed. In both procedural and functional representations, compositional constraints can be used to govern how terms match, ensuring that data types and physical quantities are combined appropriately. Composition bias in procedural representations is reflected in the use of limited repertoire of control constructs, *like if …then…else…* and *while….do…*, and even in the presuppositions of sequential execution. In future there may be greater use of parallel procedural representations, where a construct like sequence is necessary to indicate sequentially explicitly.

There is interesting relationship between conceptual and compositional bias, and formal grammars. A grammar has a terminal vocabulary, the set of all symbols in the language it generates, and set of production rules for sentence formation. To any concept space there corresponds a grammar that generates possible concepts. Conceptual biasing determines a vocabulary. It specifies only content symbols (refer to things in the domain) and not the non-content symbols such as logical connectives.

Hypothesis can not be justified deductively, they are induced from observed examples. There is no formal way to choose between competing ones, so long as they explain the examples given and can successfully predict crucial examples not in the original set.

A system entity structure is a labeled tree. Nodes of the tree are classified as entities, aspects, specializations, and multiple decompositions. Variables can be attached to nodes. They are called attached variables types. An entity signifies a conceptual part of the system being represented by the entity structure. An aspect is a mode of decomposing an entity. A specialization is a mode of classifying an entity. An entity may have several specializations or decompositions, each specialization may have several entities. The original entity is called a general type relative to the entities of a specialization. The entities of a specialization are called specialized types. Since each entity may have several specializations, a hierarchical structure called taxonomy results. A multiple decompositions is a means of representing varying number of entities. An attached variable type is an attribute of an object represented by the entity with which the variable type is associated.

Fig. 5 depicts high level view of the entity structure for chemical system building in organic chemistry. The root entity, named CHEM-1 denoted the model structure of C-carbon, H-hydrogen and O-oxygen. It has one attached variable, model constituent whose legal values are: SYSTEM CHO, SYSTEMS, CHEM-NEW and CD. Each value of the model constituent variable acts as a pointer to one of the specialized entities. The specialized entities are in turn decomposed along a segmentation aspect into k- entities, corresponding to abstract atomic segment models, such as: SUBSYSTEM1011, PS,PEPTIDE, SYSTEMA, SYSTEMI, EKB and CKB.

SUBSYSTEM 1011 means system with 10- carbon atoms, 1- oxygen, and 1-means deficit of hydrogen atoms. PEPTIDE gives cyclic and non-cyclic structure. SYSTEM generates applications for different users. SYSTEMI gives isomeric structure and numbering. SYSTEMS builds compounds which contain N- nitrogen, H-hydrogen and S-sulfur in configuration with C-carbon, H-hydrogen and O-oxygen atoms. PS gives partial structure (NMR), EKB means empirical knowledge base, CKB represents component knowledge base, and CD system for comparing data from different sources.

The system entity structure organizes a variety of system decompositions and, consequently, a variety of model constructions. Its generative capability facilities convenient definition and representation of models and their attributes at multiple levels of aggregation and abstraction.

$$X_i = g_j(f_{1j}(ROH), f_{2j}(RCOH), f_{3j}(RCOOH), \\ f_{4j}(RNH_2), f_{5j}(RSH), f_{6j}(RCPOOO)). \quad (4)$$

VIII. SYSTEM ENTITY STRUCTURE ANALYSIS

Multifacetted methodology denotes a modelling approach which recognizes the existence of multiplicities of objectives and models in any simulation project. It provides formal representation schemes that support the modeler in organizing the model construction process, in aggregating partial models, and in specifying simulation expirements. Modelling objectives drive three fundamental processes in the methodology; they facilitate the representation of model's structure, retrieval, and manipulation of structures, the specification of model's behavior, and the specification of experimental conditions under which models are evaluated by a simulation study. As a step toward a complete knowledge representation scheme for modeling support, it has combined the decomposition, taxonomic and coupling relationships

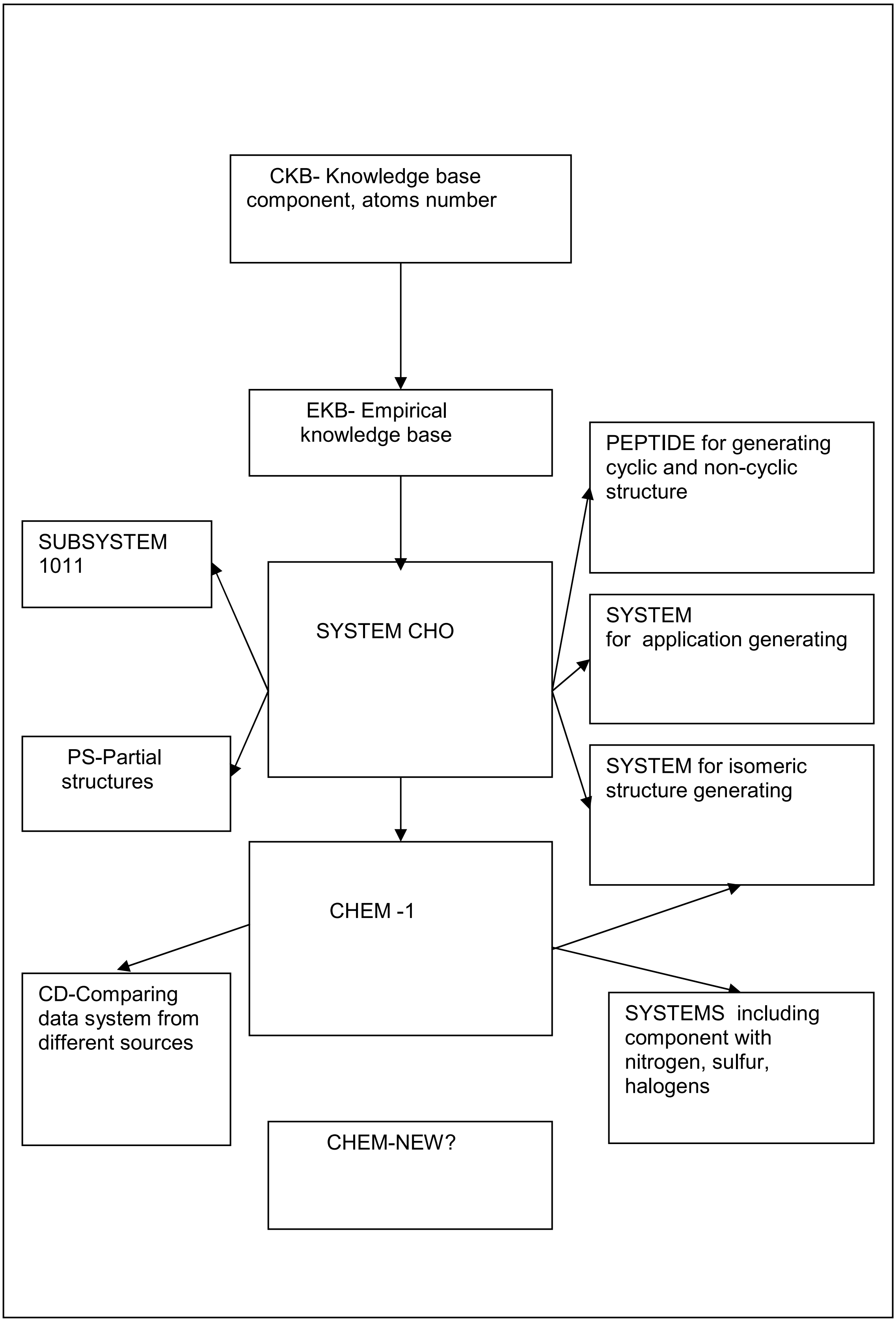

Fig.5 Organic compounds structure

in a knowledge representation scheme called the system entity structure. Previous works identified the need for representing the structure and behavior of systems, in a declarative scheme related to frame theoretical and object based formalisms. The elements represented are motivated, on the one hand, by system theory concepts of decomposition, how a system is hierarchically broken down into components and coupling how these components may be interconnected to reconstitute the original system. On the other hand, systems theory has not focused on taxonomic relations, as represented for example in frame hierarchy knowledge representation schemes. In the system entity structure scheme, such representation concerns the admissible variants of components in decompositions and the further specializations of such variants.

IX. SYSTEM CONSTRUCTION

A model is synthesized from components stored in the model base. A synthesis specification is the results of pruning a substructure from the system entity structure (Fig.6). Pruning results in a model structure candidate for a best match to the set of modelling objectives. It can be viewed as a search through the space of candidate solutions to the problem. Production rules represent the knowledge consisting of modelling objectives, coupling constraints, user requirements and performance expectations. The aim of pruning is to recommended plausible candidates for an optimal solution to the problem with respect to the requirements and constraints.

Model construction process was begin with conceptualizing decompositions and specializations of components of the system being modeled. The system entity structure base was utilized as a repository of previous modelling experience. Models associated with new atomic entities must be developed and placed in the model base.

A rule base is developed and used in the pruning process. The pruning process generates model composition trees. For each component atomic model represented by the tip node of the composition tree.

From subsystem CHEM-1 which including nitrogen and halogens can make surfactants as shown in Fig.6.

X. CONCLUSION

In this article multifacieted, knowledge based and causality modelling were studied with the aim to learne and to discovere. Models may have several submodels represented and managed by the system entity structure. The entity structure in which all components are encoded in the form of models constituentes a modelling knowledge representation scheme. Models were coupled in hierarchical way.

Deriving a simulation model from an understanding of the system to be simulated is perhaps the most complex and time consuming task of the simulation life cycle. This paper was focused for organic compounds structure formation. Model development for surfactants production was illustreted.

Simulation in data bases environment is significant simulation technique which has advantages in common parameters generating and concurrent parameters comparison.

The entity structure for chemical system building in organic chemistry was examined. The obtained results has illustrated discovering method for surfactants formation.

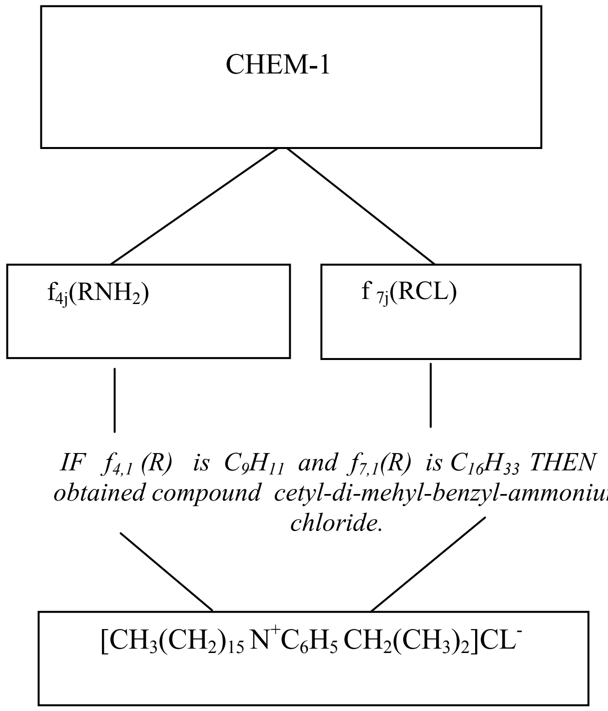

Fig.6 A surfactant construction

References:

[1] T.N. Mosorinac, An educational system in chemistry, *Computers and Ecological Engineering,* Vol.5, No.1,2009,1-6.

[2]J.B.Savkovic-Stevanovic, A system for discovering developing, *Computers and Ecological Engineering,* Vol.5, No.1,2009,7-13.

[3] T.I.Oren and B.P.Zeiger, AI in modelling and simulation direction to explore, *Simulation*,56,1988,pp.131-134.

[4] J.Savkovic-Stevanovic, An educational information communication system, *8th WSEAS Intern. Confrence on Education and Educational Technology-EDU09,*University of Genova, Genova, Italy,ID626-204, October, 17-19,2009.

[5] H.M. Jr. Blalock, *Causal Inferences in NonExperimental Research*, The Norton Library, New York,1964.

[6] M.R.Geneserth, N.J.Nilson, *Logical foundations of artificial intelligence*, Los Altos, CA, Moregan-Kaufman,1987.

[7] J.Savkovic-Stevanovic, *Informatika,* faculty of Technology and Metallurgy, Belgrade University, Belgrade, Serbia,2007.

[8] J.Savkovic-Stevanovic, Causal modeling and inference computing, *UKSIM2010, Intelligent Systems Modelling and Simulation*, Univ. of Liverpool, Liverpool, England, Jan.27-29,210.

[9] J. Savković-Stevanović, *Process engineering intelligent systems*,Srbisim, Belgrade, Serbia,2008.