Formal systems for knowledge representation are varied in their approaches. Choosing one formalism for an application typically involves tradeoffs between expressivity and tractability. For the semantic web, there is significant momentum in using description logics (DL) as a standard formalism for knowledge representation. Specifically, the W3C has proposed the Web Ontology Language (OWL) as the standard language for describing the classes and relationships of resources on the internet. The early adopters of these technologies have come from the scientific and medical communities, who have substantial requirements for knowledge representation tools and techniques.

While OWL is an expressive logic, there are classes of statements that cannot be made with the current specification. To this end, there is growing interest in the use of rules for expressing declarative knowledge that cannot be captured in the description logics used to represent ontologies on the semantic web. Rule languages enhance the expressivity of a DL ontology by providing the constructs to enable inference over explicitly represented knowledge.

A complex rule will typically contain several intermediate goals that must be satisfied before the higher order rule can be satisfied. For clarity, we will adopt terminology from the logic community and call the higher order rule a theorem, and call the intermediate goals axioms. We use this convention to distinguish elements at the knowledge-level and do not make any logical distinction between an axiom and a theorem. That is, a theorem encapsulates some chunk of domain knowledge whereas axioms provide the intermediary steps required to prove the theorem. Logically, theories can be combined to create higher order theorems using the same methods we propose for building theorems from axioms.

One common approach to modeling complex rules is to decompose the theorem into individual axioms, derive the implications of those axioms, and use the newly inferred information as an atom in a higher level axiom. This is done recursively until there is no sufficient information to continue, or the theorem is proved. For instance, suppose we have a knowledge-based system for qualitatively inferring cancer risk. One theorem in such a system may read:

If the patient has 2 or more first or second degree relatives with breast cancer risk flags then the patient has a high risk for breast cancer.

Solving this theorem requires several pieces of intermediate information. Axiom (1) must satisfy the *first and second degree relatives* pattern, to determine all the individuals in the knowledge base that satisfy this condition for the given patient. This axiom may itself be composed of lower lever axioms that infer familial relationships like aunt, uncle and cousin from the explicit information in the knowledge base. Axiom (2) must identify the *breast cancer risk flags*, which may be composed of several lower level axioms that assign breast cancer risk flags based on clinical or diagnostic information for the *first and second degree relatives* identified in by axiom (1). Axiom (1) identifies all of the relevant family members, and Axiom (2) assigns a positive breast cancer risk flag where appropriate. Only after satisfying (1) and (2) are we in a position to prove the higher order theorem, which says that if two or more of the results from (1) and (2) are positive, then we can infer that the patient of interest has a high risk of breast cancer.

This type of decomposition is typical for complex rule bases, resulting in an explosion of intermediate axioms used to generate a working context for the theorem. However, there is no formalism for knowledge engineers to explicitly model the relationship between these axioms and a theorem, or theorems and other theorems. The burden is thus on the rule author to maintain the integrity of these relationships. Incremental changes to either the rule base or the underlying ontology may have unintended consequences. Therefore, I propose to investigate a foundational model of rules for reasoning over ontologies that captures the interrelatedness of rules and the associated classes and properties in the ontology. Using concepts in an ontology as the atoms of a rule provides a powerful new layer of semantics to describe how inference rules are related. As such, these relations should be captured as first class entities in the knowledge base.

Specific Research Areas in Knowledge Management for Rules

1. ***Theorems***

   A method for describing rules (meta-rules hereafter) and a formalism for combining rules into theorems is required. An explicit representation of the dependencies of theorems on individual axioms provides a basis for understanding how additions, deletions, and modifications will impact the resulting rule-base. It will also provide the infrastructure necessary for software tool developers to create applications that guide the user through the rule authoring process. The literature on inductive logic programming contains many such resolution patterns for extrapolating and grouping related inference rules.

2. ***Rule Relations***

   Just as concepts in an ontology have properties that describe their relationships, rules also have relationships (often implicit) that define their interaction and connectedness. If we know the type of a rule we can create a graph describing how the rules interact, and use the graph to determine the effects of adding/modifying/removing a node on the overall structure of the knowledge base. This graph can be further enriched by the existing semantics of the atoms that participate in the rules, which are themselves elements of an ontology with well defined semantics. Such a representation would allow us to inspect the dependencies in a rule base, identifying the implications of modifications to both the rule base and the ontology.

   For instance, if the consequent of some rule *X* modifies one of the antecedents of another rule *Y*, this may either fulfill or negate the antecedent of *Y* depending on the context. Therefore, rule *X* and *Y* have some relationship. If we assume a closed world this relationship may be satisfiability, if it were the case that the antecedent of *Y* could **only** be satisfied by an assertion in the consequent of *X*.

3. ***Meta-rule model***

   The minimal model will describe the function of the rule – assertion, query, constraint – and the components of the rule antecedent and consequent [2]. The antecedent will describe the "inputs" to the rule; restrictions that can be represented as the domain of the rule. The consequent will describe the range of the rule and the result – modified values, new assertions, query results. The rule meta-model is necessary to define the rule relations from (2) because the type, domain and range of a rule will be used in defining the edges of the graph.

   The meta-rule model will also be used to author rules based on templates. These patterns can be safely instantiated in a particular context by mapping the concepts in the domain ontology onto the template slots of the rule using the meta-rule model to constrain the domain and range of the rule slots.