Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

# Achieving Intelligence Using Prototypes, Composition, and Analogy

**Vinay K. Chaudhri**
Artificial Intelligence Center, SRI International
Menlo Park, CA 94025, USA

**Abstract**

In this paper, I summarize the results of a decade-plus of research and development driven by the vision that human knowledge can be grounded in a small number of prototypical components that can be extended through composition and analogy. This vision has been embodied in a system called AURA, which has been used to engineer an expressive knowledge base for an intelligent biology textbook. The focus of the current paper is to abstract away from the specifics and, to instead describe the core ideas in such a manner that they can be transferred and applied in different contexts, and to relate those ideas to the ongoing research by others.

## Introduction

Our work is based on the hypothesis that, for a program to answer questions, explain the answers, and engage in a dialog just as a human does, it must have an explicit representation of knowledge (Smith 1982). A challenging task requiring such representation is capturing the knowledge found in a textbook with the goal of answering the questions that a student could answer after reading the book. Clearly, this task requires deep knowledge, because it is not just limited to answering questions, but also requires an explanation of those answers, and a conversational dialog that is sensitive to the knowledge of the two parties.

To create a program of the sort suggested above, we have been driven by the vision that knowledge can be grounded in a small number of prototypical components that can be extended through composition and analogy. We started to embody this vision in a functioning system called SHAKEN where the evaluation task was to model and reason with one page of knowledge from a biology textbook (Clark et al. 2001). This work evolved into the AURA system as part of Project Halo (Gunning et al. 2010). During 2003-2013, AURA was generalized to the domains of physics, chemistry and biology, was used to engineer a biology knowledge base (KB), and culminated with the demonstration of an intelligent textbook (Chaudhri et al. 2013a).



The project team has authored several papers describing the effort showing contributions in different sub-areas of AI. In offering my personal synthesis of the work through this paper, I have two overarching goals: (1) to summarize the central ideas that constituted the project vision providing one reference that serves as an entry into the body of literature; and (2) to share what worked and what did not in the project, to relate the work to current research by others, and to articulate some problems for future research.

AURA implementation was done by using a representation system called Knowledge Machine or KM (Clark and Porter 1999). KM offered basic support for prototypes and composition before the project started. During Project Halo, we made substantial refinements to the implementation of prototypes and composition facilities and their formalization as presented here. KM did not provide any support for analogical reasoning, and we developed special purpose problem solvers to perform such reasoning during the project.

This paper includes three major sections:(1) prototypes, (2) composition and (3) analogy. I begin each section with a definition, then discuss a realization in the context of AURA and finally follow with a discussion of the successes and challenges, related work, and problems open for future research. Prototypes, composition and analogy are big long-term problems, and I do not claim that they have been solved completely. My focus is on explaining what was accomplished during the project. Towards the the paper's end, I describe the KB construction undertaken using AURA, an end-user application built by using the KB and the challenges faced in transitioning it into a commercial setting.

## Prototypes

Prototype theory is a mode of graded categorization in cognitive science, where some members of a category are more central than others (Rosch 1983). For example, when asked to give an example of the concept furniture, chair is more frequently cited than, say, stool. A prototype of a class is the most salient member of that class.

### Formalizing Prototypes

We reduced the idea of capturing the most salient member of a class to capturing universally true properties for the instances of that class. We illustrate this approach by giving an example. Suppose we wish to represent the statement: "*Every cell has part a chromosome and a ribosome*." Given a class Living-Entity, we can represent this knowledge in first order logic as follows:

$$\begin{aligned}&\forall x[Cell(x) \Rightarrow \\ &\quad \exists r,c[Living\text{-}Entity(x) \wedge Ribosome(r) \wedge \\ &\qquad Chromosome(c) \wedge has\text{-}part(x,c) \wedge \\ &\qquad has\text{-}part(x,r)]]\end{aligned} \tag{1}$$

Next, consider the first order logic representation of the statement: "*Every eukaryotic cell is a cell and has part a nucleus and a eukaryotic chromosome such that the eukaryotic chromosome is inside the nucleus*."

$$\begin{aligned}&\forall x[Eukaryotic\text{-}Cell(x) \Rightarrow \\ &\quad \exists n,c[Cell(x) \wedge Nucleus(n) \wedge \\ &\qquad Eukaryotic\text{-}Chromosome(c) \wedge has\text{-}part(x,c) \wedge \\ &\qquad has\text{-}part(x,n) \wedge is\text{-}inside(c,n)]]\end{aligned} \tag{2}$$

In rule 1 (rule 2), we capture prototypical instance of a cell (eukaryotic cell). Exceptions may exist (for example, situations where chromosomes have been removed from a cell) in which these rules are not true. The focus in the project was capturing what could be considered universally true for the salient instances of a class (i.e., universally true or prototypical properties of that class using rules such as 1 and 2.)

Both of the above rules have a form in which the antecedent is universally quantified and the consequent is existentially quantified. Such rules have been referred to as existential rules in the literature (Baget et al. 2011). Each prototype in AURA is captured by using such existential rules. A more detailed formalization of the representation in AURA is available elsewhere (Chaudhri et al. 2013b).

## Successes and Challenges

A major success in using prototypes as formalized in the previous section was that they also provided a basis for an intuitive user interface for knowledge acquisition from domain experts (Chaudhri et al. 2007). This graphical interface had extensions that dealt with more expressive forms of knowledge such as constraints, sufficient properties, mathematical equations and tables (Chaudhri et al. 2004).

A major challenge in using prototypes in AURA was that their formal properties were not well understood. The prototypes were organized in a class hierarchy, they could inherit values across a class hierarchy, participate in multiple inheritance, and participate in circular references to each other. To understand the relationship of prototypes to the current state of KR research, let us compare them with description logics. For example, rule 2 can be captured by using DL syntax (Baader, Horrocks, and Sattler 2008) as follows:

$$\begin{aligned}&Eukaryotic\text{-}Cell \equiv Cell \sqcap \\ &\quad (\exists has\text{-}part.Nucleus) \sqcap \\ &\qquad (\exists has\text{-}part.(Eukaryotic\text{-}Chromosome \sqcap \\ &\qquad\quad (\exists is\text{-}inside.Nucleus)))\end{aligned} \tag{3}$$

The above description fails to represent that the eukaryotic chromosome is inside the same nucleus that is a part of the eukaryotic cell. This problem cannot be solved by introducing qualified number constraints on the has-part relation for Chromosome, Eukaryotic-Chromsome and Nucleus, because they are too strict. Expressing such knowledge requires violating the desirable tree model property (in general, the tree model property is a good indicator of decidability (Vardi 1996)). Reasoning with prototypes is an undecidable problem. In AURA, we dealt with this issue by setting a bound on the depth of inference performed, thus, trading completeness for efficiency. The solution is only partially effective as numerous questions took as much as 30 minutes to answer, and many questions failed to answer even when the requisite knowledge was present in the KB.

## Future Research

The prototypes in AURA do not support the notion of graded membership as originally introduced (Rosch 1983). This limits AURA from capturing exceptions to the prototypical case that are so frequent in biology. Leveraging the notion of graded membership is a possible strategy to combine a prototype-based representation with the corpus-based statistical methods that can suggest how frequently a particular instance of a concept occurs. Many forms of knowledge in textbooks do not naturally fit into the structure of an existential rule (for example, computational procedures, algorithms, mathematical equations, etc.).

Several efforts are underway to address efficient reasoning with the existential rules needed for representing prototypes (Magka, Krötzsch, and Horrocks 2013; Eiter and Šimkus 2010; Calì, Gottlob, and Lukasiewicz 2009; Alviano, Faber, and Leone 2010). All of these efforts achieve decidable reasoning by imposing restrictions on the syntactic structure of the KB. For example, one such notion is called R-acylicity (Magka, Krötzsch, and Horrocks 2013). We say that a rule $r_2$ positively relies on another rule $r_1$ if a situation exists (represented by a set of facts) where $r_1$ is applicable, $r_2$ is not applicable, and applying $r_1$ allows $r_2$ to derive something new. Consider the following two rules:

$$A(x) \Rightarrow \exists y[B(y), r(x,y)] \tag{4}$$

$$B(x) \Rightarrow C(x) \tag{5}$$

Consider a set of facts $\{A(a)\}$ such that rule 4 is applicable to those facts (i.e., $A(a)$ holds) and rule 5 is not applicable to those facts as no $B$ atom is present. Applying 4 gives us some $b$ such that $r(a,b), B(b)$ holds, which allows 5 to derive something new namely $C(b)$. Rule 5 relies on 4 to derive a conclusion. A KB is R-acyclic if no cycles of positive reliances exist that involve rules with existential quantifiers in the head. Many examples of textbook knowledge violate R-acyclicity as seen in the example below.

$$\begin{aligned}&Peripheral\text{-}Protein(x) \Rightarrow \\ &\quad \exists b[Membrane\text{-}Surface(b), is\text{-}attached\text{-}to(x,b)]\end{aligned} \tag{6}$$

$$\begin{aligned}&Membrane\text{-}Surface(x) \Rightarrow \\ &\quad \exists m[Membrane(m), has\text{-}region(m,x)]\end{aligned} \tag{7}$$

$$\begin{aligned}&Membrane(x) \Rightarrow \\ &\quad \exists p[Peripheral\text{-}Protein(p), has\text{-}part(x,p)]\end{aligned} \tag{8}$$

Here rule 6 positively relies on rule 8, rule 8 relies on rule 7, and rule 7 relies on rule 6, and thus, violates R-acyclicity. We need further research on efficient reasoning with such KBs. There is also a need to define update semantics when the KB contains prototypes that refer to each other in circular ways.

## Composition

The principle of compositionality says that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. The simplest form of composition is through properties (for example, "tall person", "red car", etc.). Next, one can construct complex expressions by associating values to relations (for example, "animal respiration" is a respiration that is performed by an animal). Finally, one can compose new expressions through inheritance. For example, "eukaryotic aerobic respiration" is a respiration that specializes both "aerobic respiration" and "eukaryotic respiration".

### Formalizing Composition

AURA supports a novel solution for composition for inheritance reasoning in an under-specified KB (Chaudhri and Tran 2012). A KB is under-specified, if it omits some cardinality constraints and equality statements. In the example considered earlier, Eukaryotic-Cell inherits a Chromosome from its super class Cell which is then specialized to Eukaryotic-Chromosome. Rules 1 and 2 do not explicitly state the relationship between the inherited Chromosome and the Eukaryotic-Chromosome. We can address that by rewriting them as follows:

$$\begin{aligned}&\forall x[\mathit{Cell}(x) \Rightarrow\\ &\quad[\mathit{Living\text{-}Entity}(x) \wedge \mathit{Ribosome}(f_1(x))\wedge\\ &\qquad \mathit{Chromosome}(f_2(x)) \wedge \mathit{has\text{-}part}(x, f_1(x))\wedge\\ &\qquad \mathit{has\text{-}part}(x, f_2(x))]\end{aligned} \tag{9}$$

$$\begin{aligned}&\forall x[\mathit{Eukaryotic\text{-}Cell}(x) \Leftrightarrow\\ &\qquad \mathit{Cell}(x) \wedge \mathit{Nucleus}(f_3(x))\wedge\\ &\qquad \mathit{Eukaryotic\text{-}Chromosome}(f_4(x))\wedge\\ &\qquad \mathit{has\text{-}part}(x, f_3(x)) \wedge \mathit{has\text{-}part}(x, f_4(x))\wedge\\ &\qquad \mathit{is\text{-}inside}(f_3(x), f_4(x))]\end{aligned} \tag{10}$$

$$\begin{aligned}&\forall x[\mathit{Eukaryotic\text{-}Cell}(x) \Rightarrow\\ &\quad f_2(x) = f_4(x) \wedge f_1(x) = f_3(x)]\end{aligned} \tag{11}$$

Rules 9-11 enable capturing the inheritance relationships across a class hierarchy. These rules would be under-specified if rule 11 were to be omitted from the KB. Two motivations exist in AURA for allowing such under-specification: (1) knowledge authoring is much easier if the encoder does not need to specify the equality statements; and (2) un-anticipated multiple inheritance may exist, and specifying all such equality statements ahead of time is infeasible. For example, a class such as eukaryotic aerobic respiration may not pre-exist in the KB, and may need to be computed only at run time. If the knowledge was to be acquired from natural language text, under-specification is rampant, and leaving the representation under-specified, and inferring the equalities at run time is preferable.

AURA supports heuristic inference to compute such equalities at run time. This inference is performed when an instance of a class inherits values from one or more super classes, or during knowledge authoring when the user asserts two instances to be equal. To perform inheritance from multiple super classes, AURA first creates an instance of that class and each of its super classes and composes (i.e., equates) them. The intuition behind heuristic composition is as follows: if a slot has only one value of each type, for each of the two objects being composed, then equate them; if the slot has two values one of which is more specific than the other, then the two values are equated; if a slot has more than one value of each type, but they can be distinguished based on certain properties, equate them appropriately. In the example above, because both Cell and Eukaryotic-Cell have only one value of type Ribosome, we can conclude $f_1(x) = f_3(x)$. Because Eukaryotic-Chromosome is more specific than Chromosome, we can conclude $f_2(x) = f_4(x)$. The heuristic inference leverages any inferences that follow deductively, and does not make any inferences that are disallowed. For example, two values for a functional slot will be considered equal and two instances of disjoint classes will never be considered equal.

### Successes and Challenges

Composition is effective as long as each slot has only one value of a particular type, or if multiple values of the same type exist, they can be distinguished. Such reasoning breaks down whenever slot values exist that are indeed different but cannot be distinguished. We illustrate this with an example. We have three classes: Amino-Acid, Polar-Molecule and Polar-Amino-Acid. The Polar-Amino-Acid is a subclass of both Amino-Acid and Polar-Molecule, thus, introducing multiple inheritance. Next, we will introduce snippets of prototype rules for these concepts.

$$\begin{aligned}&\forall x[\mathit{instance\text{-}of}(x, \textsf{Polar-Molecule}) \Rightarrow\\ &\quad \exists a1, a2, p[\\ &\qquad \mathit{instance\text{-}of}(p, \textsf{Polar-Covalent-Bond})\wedge\\ &\qquad \mathit{instance\text{-}of}(a1, \textsf{Atom})\wedge\\ &\qquad \mathit{instance\text{-}of}(a2, \textsf{Atom})\wedge\\ &\qquad \mathit{possesses}(x, p)\wedge\\ &\qquad \mathit{has\text{-}part}(x, a1) \wedge \mathit{has\text{-}part}(x, a2)\\ &\qquad \mathit{is\text{-}between}(p, a1, a2)]]\end{aligned} \tag{12}$$

$$\begin{aligned}&\forall x[\mathit{instance\text{-}of}(x, \textsf{Amino-Acid}) \Rightarrow\\ &\quad \exists h, n, c, o, b1, b2.[\\ &\qquad \mathit{instance\text{-}of}(b1, \textsf{Single-Bond})\wedge\\ &\qquad \mathit{instance\text{-}of}(b2, \textsf{Double-Bond})\wedge\\ &\qquad \mathit{instance\text{-}of}(h, \textsf{Hydrogen})\wedge\\ &\qquad \mathit{instance\text{-}of}(n, \textsf{Nitrogen})\wedge\\ &\qquad \mathit{instance\text{-}of}(c, \textsf{Carbon})\wedge\\ &\qquad \mathit{instance\text{-}of}(o, \textsf{Oxygen})\wedge\\ &\qquad \mathit{possesses}(x, b1) \wedge \mathit{possesses}(x, b2)\wedge\\ &\qquad \mathit{has\text{-}part}(x, h) \wedge \mathit{has\text{-}part}(x, n)\\ &\qquad \mathit{has\text{-}part}(x, o) \wedge \mathit{has\text{-}part}(x, c)\\ &\qquad \mathit{is\text{-}between}(b1, h, n) \wedge \mathit{is\text{-}between}(b2, c, o)]]\end{aligned} \tag{13}$$

Rule 12 states that a Polar-Molecule has two atoms as its parts and possesses a Polar-Covalent-Bond that is between those two atoms. Rule 13 states that an Amino-Acid has parts Hydrogen, Oxygen, Nitrogen and Carbon, and that it possesses a Single-Bond between the Hydrogen and the Nitrogen, and a Double-Bond between a Carbon and Oxygen. Because a Polar-Amino-Acid is a subclass of both Polar-Molecule and Amino-Acid, it will inherit two atoms from Polar-Molecule and four atoms (a Carbon, a Hydrogen, a Oxygen and a Nitrogen) from Amino-Acid. Further, it will inherit a Polar-Covalent-Bond from Polar-Molecule and a Single-Bond and a Double-Bond from the Amino-Acid. Because Polar-Covalent-Bond, Single-Bond, and Double-Bond are all sibling classes in the class hierarchy, they could not all be referring to the same bond, and therefore, they will be inherited as separate individuals. But, because Carbon, Hydrogen, Oxygen and Nitrogen are all subclasses of Atom, the two atoms inherited from the Polar-Molecule could possibly be referring to two of the four atoms being inherited from Amino-Acid. In the current system, two of the atoms inherited from Polar-Molecule would get equated with two of the atoms from Polar-Amino-Acid giving an incorrect behavior. The domain experts observed this behavior, as they were unable to override the automatic conclusions derived by the system. In many other examples, such ambiguous multiple inheritance leads to violations of the integrity constraints in the system which are reported as errors by the inference engine. One can add additional rules for such inferences to work correctly. For example, in this case one could add a default rule that prevents atoms from participating in two different bonds, but that will conflict with the Chemistry knowledge. Getting such default rules to work well across a variety of situations is extremely difficult in practice.

### Future Research

Better techniques need to be developed to deal with cases of ambiguous multiple-inheritance. One possible approach is to provide greater control to users so that they can provide guidance on how to resolve the ambiguities. Another possible direction is combining the heuristics with the statistics derived from the textbook to determine whether two entities should be equated. The view of the composition used in AURA was purely logical. The recent developments that can learn conventional aspects of natural language meaning from corpora and databases create tremendous opportunity to seek synergies between the two approaches (Liang and Potts 2015). For example, one specific challenge that could be attempted by using such a combined technique is understanding the noun phrases and verb phrases. The textbook content could provide the data needed to learn the different combinations of words that can occur together, which could then be used to learn the rules of composition.

## Analogy

Analogical reasoning and similarity reasoning both rely on an alignment of relational structure. But, they differ such that in analogy, only relational predicates are shared; whereas in literal similarity, both relational predicates and object attributes are shared (Gentner and Markman 1997). As an example: a comparison between an atom and a solar system is considered an analogy, but a comparison between a red door with a red key and a blue door with a blue key is considered a similarity. It has also been argued that the comparison process involves a sophisticated process of structural alignment and mapping over rich complex representations (Falkenhainer, Forbus, and Gentner 1989).

### Formalizing Analogical Reasoning

In AURA, two question formats support analogical reasoning: (1) What is the similarity/difference between concept *A* and concept *B*?; and (2) A concept *A* is to concept *B* as the concept *C* is to what? Example instantiations of these questions are: (1) What are the differences between an aldehyde and an alcohol?; and (2) A chloroplast is to photosynthesis as mitochondrion is to what? Both of these questions require different computations which we explain next.

The approach to answering a similarities and differences question in AURA follows the conventional model of analogical reasoning (Nicholson and Forbus 2002): case construction, candidate inference computation and summarization. A case for a concept contains its taxonomic information, slot values, and constraints. A novel aspect of case construction in AURA is including only that information in a case that is *local* to a class (i.e., cannot be derived if the rules for that class were to be dropped from the KB). The candidate inference computation is formalized as a set-theoretic computation over two case descriptions: the difference contains a slot value triple if (1) it contains a non-Skolem value that appears in one case but not in another; (2) if it contains a slot value that cannot be paired with any slot value for the same slot across the two cases. Two values can be paired (2a) if they have exactly the same types in both case descriptions, or if the types of the value in one subsume the types of the value in the other; (2b) if the values have further slot values, then the comparison must be recursively performed. The difference contains a constraint value if (1) the same constraint does not appear in both descriptions, and (2) the constraint in one cannot be derived from some constraints in the other. After the difference is computed, the results are grouped and organized into a table. The grouping is done based on the slot hierarchy in the KB and user preferences. The alignment is done based on syntactic (e.g., nucleus will be aligned with nucleolus), and semantic (e.g., sibling concepts in a taxonomy will be aligned) criteria. A more detailed description of our approach is available elsewhere (Chaudhri et al. 2014). As an example, when asked to compare an aldehyde and alcohol, the system returns an answer saying: "An aldehyde has a carbonyl group as a subunit while an alcohol has a hydroxyl group as a subunit. Alcohol has hydrogen and oxygen as subunits and possesses covalent bonds."

Let us now consider the question format: "Concept *A* is to concept *B* as concept *C* is to what?" Reasoning to answer questions of this form involves first computing a path connecting *A* and *B*, and then searching the whole KB for the same path between *C* and some *D* (Chaudhri, Dinesh, and Heller 2013). Because arbitrary search can be computationally expensive, we prioritize the search process by using the following heuristics: (1) Look for a taxonomic path between *A* and *B*, and if found, look for a similar taxonomic path between *C* and some *D*; (2) look for a path between *A* and *B* in the concept definition of *A*, and then look for the same path in the concept definition for *C*; (3) look for a path between *A* and *B* in any single concept definition in the KB, and then look for the same path between *C* and some entity *D* in any concept definition in the KB; (4) search the whole KB for a path between *A* and *B* and then look for the same path between *C* and some *D*. Such reasoning returns multiple answers, and the system sorts and ranks them. More advanced forms of similarity reasoning could relax the requirement that the paths between *A* and *B* must be identical to the path between *C* and *D*. As an example, when asked the question: A chloroplast is to photosynthesis as mitochondrion is to what, the system returns two answers: "Cellular Respiration and Chemiosmosis", and indicates that they are both functions of mitochondrion just like photosynthesis is a function of a chloroplast.

### Successes and Challenges

AURA implements both forms of the analogical reasoning presented in the previous section. The user feedback on such questions was extremely positive. Let us next consider some challenges that we encountered.

When information about two concepts is put next to each other as in the answer of a comparison question, it can create surprises. An empty entry in a comparison table is an instance of this problem. In some cases, an empty entry may be seen because the textbook does not mention a fact. For example, when we compare a ribosome and a chromosome, the system responds by saying that the function of a ribosome is protein synthesis, but the corresponding function for a chromosome is left empty. The textbook does not explicitly offer that information, and therefore, it is not present in the KB. Such textbook deficiency is rarely caught and is easily forgiven by human readers, but when it is noticed through comparison, the user feedback is extremely harsh. This observation implies that comparison questions can serve as a powerful tool during knowledge engineering to test and debug the KB as well as to improve the textbook.

For the question "Concept *A* is to concept *B* as concept *C* is to what?", a major challenge was how to best present the answer. The current system shows the answer graphically which created much confusion among the users. A better presentation would be to show such answers in English text which requires further research on natural language generation methods (Banik, Gardent, and Kow 2013).

### Future Research

We begin by situating our work in the broader context of analogical thinking (Keane, Ledgeway, and Duff 1994) by acknowledging that we only focus on the representation and mapping steps, and to a limited degree on induction. Our mapping step differs from a general model of analogical mapping in that only those concepts that have a common relation are matched. For example, a Nucleus will be matched to Nucleoid only if both are in a has-part relationship to the two cells being compared. In general analogical reasoning, the two concepts could be matched even if they are in different relationships to the concepts being compared.

There has been related work to formalize the comparison reasoning (Baral and Liang 2012). Because their work was not accompanied by any functional implementation that could be rated by end-users, it missed out central challenge of identifying the most salient similarities and differences. Our work addressed those challenges through innovations in case construction, summarization and ranking.

Answering analogical reasoning questions by using a purely textual approach (Liu, Wagner, and Birnbaum 2007) is possible. The main advantage of a purely textual approach is it does not require any KB curation. But, a significant disadvantage of a textual approach is the text may not contain sentences that explicitly compare the two concepts. A fruitful direction for future research is exploring algorithms that can exploit a combination of the two approaches.

More broadly, I believe that the analgoical reasoning in the form of the comparison and relationship reasoning questions considered here is an unconquered frontiers for AI research. For example, one can ask more specific forms of comparisons such as: What are the structural differences between *A* and *B*? Human-authored answers that compare *A* and *B* may exist, but we do not expect all specific forms of comparisons (e.g., structural, functional, regulatory, evolutionary, etc.) to be have been anticipated in advanced and pre-authored by humans.

## Knowledge Base Construction

To support knowledge base construction, AURA incorporated a library of pre-built representations called the Component Library or CLIB which the domain experts could use to create domain-specific knowledge (Barker, Porter, and Clark 2001). CLIB was substantially extended during the project to represent knowledge about biology core themes such as structure and function (Chaudhri, Dinesh, and Heller 2013), process regulation and energy transfer (Chaudhri, Dinesh, and Heymans 2014).

We used AURA to create a KB called `KB_Bio_101` that encodes a significant fraction of a biology textbook (Reece et al. 2011). `KB_Bio_101` contains more than 6000 classes, and more than 100,000 rules. To create `KB_Bio_101`, a team that largely consisted of biologists and some knowledge engineers, relied on a knowledge factory process and a set of guidelines that specified a systematic process to encode textbook sentences (Chaudhri, Dinesh, and Inclezan 2014). The team was based in India at an organization called Evalueserve, and thus, was at an arms length from the research team that created AURA. The AURA team trained two knowledge engineers from Evalueserve, who were in turn, responsible for training the biologists based in India. At its peak, the knowledge encoding team consisted of eight biologists and two knowledge engineers. The biologists had no prior training in knowledge representation. They were selected through an aptitude test designed to test their conceptual abilities, and were put through a 20 hour AURA training. `KB_Bio_101` is the first attempt to put this level of expressiveness in the hands of biologists, and also the first effort to use an off-shore knowledge entry team.

To evaluate the KB, we assembled a test suite of 1961 questions spread over the first eleven chapters of the textbook. These questions contained instantiations of several different question formats including questions that used analogical reasoning (Chaudhri et al. 2014; Chaudhri, Dinesh, and Heller 2013). Biologist scored 85% of the answer outputs as correct.

## Application to an Intelligent Textbook

To put the technology developed here in the hands of end-users, we incorporated KB_Bio_101 and AURA's question answering capability into a prototype of an intelligent textbook called *Inquire* (Chaudhri et al. 2013a). As the students read the textbook using *Inquire*, they can ask questions by typing free-form natural language queries or by selecting passages of text. The system then attempts to answer the question and also generates suggested questions related to the query or selection. The questions supported by the system were chosen to be educationally useful (for example: What is the structure of *X*?; What are the similarities and differences between *X* and *Y*?; How does *X* relate to *Y*? *A* is to *B* as *C* is to what?).

We evaluated *Inquire* with community college students. There were three conditions: (1) paper textbook (N=23), (2) a traditional electronic textbook (N=25), and (3) full-featured *Inquire* (N=24). The students in each group engaged in active reading and problem solving on the topic of membrane structure and function. The problem-solving task was comparable to what students might do during an assigned homework. During the evaluation exercise, the students worked on problem solving in an open book format, and at the end of the exercise, answered quiz questions in a closed-book format. We compared the problem solving and quiz scores across the three groups. The quiz scores of students using *Inquire* were higher than the scores of the students using either the paper textbook (p value=0.05) or electronic textbook by approximately 10% (p value=0.002). The observed trend is consistent with our hypothesis that *Inquire* enhances student learning.

Although the results presented here are encouraging, they are only preliminary. Whether these results will generalize if the students were to study from *Inquire* over an extended period of time is unknown. We also observed that the students in the *Inquire* group did not get any Ds or Fs suggesting that it may be especially helpful for lower performing students. This result, however, was not statistically significant, and more extensive experimentation is needed to confirm it.

## Transition Challenges

A major challenge we have faced in translating this success into a commercial enterprise is the cost of creating KB_Bio_101 was still too high. Some of the reasons for this high cost are as follows. The theoretical problems in reasoning with prototypes are not well-understood, making reasoning performance too slow and unpredictable. Also, the current implementation of the composition operation in KM makes destructive updates to the KB (i.e., as the system derives new information, it overwrites user-authored knowledge.). This problem was not an issue while the KB was small, but as we tried to scale up the knowledge entry, it led to too much wasted effort. The current implementation of composition also provides no user control making correcting erroneous conclusions derived by the system difficult. Our current system does an adequate job of capturing explicitly stated knowledge, but relatively little is known about how to capture knowledge about salience in analogical reasoning. To fully and deeply capture the knowledge found in a textbook, continued work on expanding the component library to provide domain independent representations is needed. Finally, recognizing that the problems articulated here are long-term research problems in AI, and that intermediate impactful outcomes are needed, a clear need exists for product-focused R&D that can package the already available capability into a minimum viable product.

## Summary

In this paper, I presented an approach for realizing intelligence that is based on the notion that knowledge should be grounded in a small number of prototypical components that could be reasoned with by using composition and analogy. This approach has been embodied in a system called AURA. Using prototypes was central in AURA to focus the knowledge acquisition to universal truths, and to provide a basis for designing a user interface for domain experts. Prototypes in AURA can be formalized as graph-structured existential rules, which is a topic for ongoing research in both description logics and logic programming. AURA realized the principle of compositionality by inheritance reasoning in underspecified knowledge bases. Compositionality is of great interest for the current research on natural language understanding, and the lessons learned from its implementation in AURA can provide new inspirations for combining logical methods with corpus-based methods. AURA used two innovative implementations for analogical reasoning and generated answers of sophistication that surpass any available question answering system. AURA was used for large scale knowledge engineering the results of which were incorporated into an intelligent textbook for students. Although this prototype may not be ready for commercial use, it provides a platform for innovations in online learning that promise not just to transform education, but also to provide a concrete focus for further research in achieving intelligence through prototypes, composition and analogy.

## Acknowledgments

This work has been funded by Vulcan Inc. and SRI International. I gratefully acknowledge the contributions of Peter Clark, Paul Cohen and Bruce Porter who were responsible for formulating the vision presented here. I also acknowledge the support of Mark Greaves, Benjamin Grosof and Dave Gunning for promoting the project at Vulcan. Finally, I thank the KB development team, the biology teachers and the software engineering team who contributed to this effort.

# References


Alviano, M.; Faber, W.; and Leone, N. 2010. Disjunctive ASP with functions: Decidable queries and effective computation. *Theory and Practice of Logic Programming* 10(4-6):497–512.

Baader, F.; Horrocks, I.; and Sattler, U. 2008. Description logics. *Handbook of Knowledge Representation and Reasoning* 3:135–179.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2011. On rules with existential variables: Walking the decidability line. *Artificial Intelligence* 175(9):1620–1654.

Banik, E.; Gardent, C.; and Kow, E. 2013. The KBGen challenge. In *ENLG 2013 : 14th European Workshop on Natural Language Generation*.

Baral, C., and Liang, S. 2012. From knowledge represented in frame-based languages to declarative representation and reasoning via ASP. In *Proceedings of the Interntational Conference on Knowledge Representation and Reasoning*.

Barker, K.; Porter, B.; and Clark, P. 2001. A Library of Generic Concepts for Composing Knowledge Bases. In *First International Conference on Knowledge Capture*.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2009. Datalog$\pm$: A unified approach to ontologies and integrity constraints. In *Proceedings of the 12th International Conference on Database Theory*, 14–30. ACM.

Chaudhri, V. K., and Tran, S. C. 2012. Specifying and reasoning with under-specified knowledge base. In *International Conference on Knowledge Representation and Reasoning*.

Chaudhri, V.; Murray, K.; Pacheco, J.; Clark, P.; Porter, B.; and Hayes, P. 2004. Graph-based acquisition of expressive knowledge. In *Proceedings of the International Conference on Knowledge Engineering and Management*.

Chaudhri, V.; John, B. E.; Mishra, S.; Pacheco, J.; Porter, B.; and Spaulding, A. 2007. Enabling experts to build knowledge bases from science textbooks. In *The Proceedings of the International Conference on Knowledge Capture*.

Chaudhri, V. K.; Cheng, B.; Overholtzer, A.; Roschelle, J.; Spaulding, A.; Clark, P.; Greaves, M.; and Gunning, D. 2013a. *Inquire Biology*: A textbook that answers questions. *AI Magazine* 34(3).

Chaudhri, V. K.; Heymans, S.; Wessel, M.; and Tran, S. C. 2013b. Object-oriented knowledge bases in logic programming. In *Technical Communication of International Conference in Logic Programming*.

Chaudhri, V. K.; Heymans, S.; Overholtzer, A.; Spaulding, A.; and Wessel, M. 2014. Large-scale analogical reasoning. In *Proceedings of AAAI-2014 Conference*.

Chaudhri, V. K.; Dinesh, N.; and Heller, C. 2013. Conceptual models of structure and function. In *Second Annual Conference on Advances in Cognitive Systems*.

Chaudhri, V. K.; Dinesh, N.; and Heymans, S. 2014. Conceptual models of energy transfer and regulation. In *Proceedings of International Conference on Formal Ontologies in Information Systems*.

Chaudhri, V. K.; Dinesh, N.; and Inclezan, D. 2014. Creating a knowledge base to enable explanation, reasoning, and dialog. *Advances in Cognitive Systems* 3:183–200.

Clark, P., and Porter, B. 1999. KM – The knowledge machine: Users manual. Technical report, `http://www.cs.utexas.edu/users/mfkb/`.

Clark, P.; Thompson, J.; Barker, K.; Porter, B.; Chaudhri, V.; Rodriguez, A.; Thomere, J.; Mishra, S.; Gil, Y.; Hayes, P.; and Reichherzer, T. 2001. Knowledge entry as the graphical assembly of components. In *First International Conference on Knowledge Capture*.

Eiter, T., and Šimkus, M. 2010. $\mathbb{FDNC}$: Decidable non-monotonic disjunctive logic programs with function symbols. *ACM Transactions on Computational Logic* 11(2):14.

Falkenhainer, B.; Forbus, K.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41:1–63.

Gentner, D., and Markman, A. B. 1997. Structure mapping in analogy and similarity. *American Psychologist* 52:45–56.

Gunning, D.; Chaudhri, V. K.; Clark, P.; Barker, K.; Chaw, S.-Y.; Greaves, M.; Grosof, B.; Leung, A.; McDonald, D.; Mishra, S.; Pacheco, J.; Porter, B.; Spaulding, A.; Tecuci, D.; and Tien, J. 2010. Project Halo update: Progress toward digital Aristotle. *AI Magazine*.

Keane, M. T.; Ledgeway, T.; and Duff, S. 1994. Constraints on analogical mapping: A comparison of three models. *Cognitive Science* 18(3):387–438.

Liang, P., and Potts, C. 2015. Corpus-based semantics and pragmatics. *Annual Review of Linguistics*.

Liu, J.; Wagner, E.; and Birnbaum, L. 2007. Compare&contrast: Using the web to discover comparable cases for news stories. In *Proceedings of the 16th international conference on World Wide Web*, 541–550. ACM.

Magka, D.; Krötzsch, M.; and Horrocks, I. 2013. Computing stable models for nonmonotonic existential rules. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*.

Nicholson, S., and Forbus, K. D. 2002. Answering comparison questions in SHAKEN: A progress report. In *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Reece, J. B.; Urry, L. A.; Cain, M. L.; Wasserman, S. A.; Minorsky, P. V.; and Jackson, R. B. 2011. *Campbell biology*. Boston: Benjamin Cummings imprint of Pearson.

Rosch, E. 1983. Prototype classification and logical classification: The two systems. *New trends in conceptual representation: Challenges to Piagets theory* 73–86.

Smith, B. C. 1982. *Reflection and Semantics in a Procedural Language*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Vardi, M. Y. 1996. Why is modal logic so robustly decidable? *Descriptive Complexity and Finite Models* 31:149–184.