Representing States in a Biology Textbook

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

Daniela Inclezan
Department of Computer Science and
Software Engineering, Miami University
Oxford, OH 45056

Abstract

Representing biology textbook knowledge involves handling numerous concepts that have multiple possible states, for example, developmental states such as embryo, juvenile and larva; system states such as homeostasis and equilibrium; states of chromosomes such as chromatin, nicked, etc. Though substantial research exists on formalisms for representing states, relatively less work exists on ontologically representing them in a complex domain. Our findings include: (a) the word *state* in natural language is used with both entities and events which requires that we generalize the traditional definition of state to distinguish between an entity state and an event state; (b) an abstract modeling pattern called the *process flow diagram* that provides a practically achievable target for the output of natural language processing programs, and enables knowledge authoring by domain experts that can be compiled into a well-known background theory based on action languages. The background theory, combined with reasoning methods from the action language, allows building tools that simulate processes and answer sophisticated questions about process interruptions.

Introduction

For Project Halo, we recently completed a knowledge-engineering effort that resulted in a knowledge base called `KB_Bio_101` (Chaudhri et al. 2014), which represents a significant fraction of an introductory college-level biology textbook (Reece et al. 2011). We have used `KB_Bio_101` in a prototype of an intelligent textbook called Inquire, which is designed to help students to learn better (Chaudhri et al. 2013a). Inquire answers questions (Chaudhri et al. 2013b), gives explanations and engages in dialog through natural language generation (Banik, Kow, and Chaudhri 2013).

The biology textbook contains several concepts that have a flavor of states (for example, developmental states, states of reactions, states of molecules, and states of a complex dynamic system). As we can see, the textbook uses the word *state* to refer to both events (e.g., Homeostasis, Equilibrium, etc.) and entities (e.g., Blastula, Chromatin, etc.). Most upper ontologies define Event and Entity as disjoint classes (Gangemi et al. 2003; Niles and Pease 2001; Lenat 1995). Previous work on representing states (McCarthy and Hayes 1968; Reiter 2001; Henzinger 2000) does not address how to represent states in a complex domain when they could be either entities or events. We have, therefore, structured our work to provide clear answers to the following questions: (1) what counts as a state; (2) where to put state concepts in a taxonomy; (3) how the state concepts should be used in the knowledge representation. The answer to the first question addresses the usage of the word state for both entities and events. The answer to the second question clarifies the relationship of the state concepts for entities and events to an upper ontology. The answer to the third question provides a representation that can be used as the target representation by the natural language processing (NLP) programs as well as by subject matter experts.

Biological concepts with a flavor of states are closely related to the concepts of an action and a process which we use interchangeably in this paper. Representing processes has been an active area of interest in knowledge representation and reasoning (KRR), upper ontologies, and NLP. KRR researchers have developed a variety of action representation languages that can be used for modeling processes (Gelfond and Lifschitz 1993; Baral 2003). Most upper ontologies such as DOLCE (Gangemi et al. 2003), SUMO (Niles and Pease 2001), and Cyc (Lenat 1995) support Event (or comparable concepts like Occurrent or Perdurant) as a distinction in the ontology, and a variety of associated relationships. The NLP community has developed case roles and created event lexicons such as VerbNet (Kipper et al. 2008) and FrameNet (Baker, Fillmore, and Lowe 1998).

The contributions of work are to connect the usage of the word *state* to appropriate constructs in the KRR language and an upper ontology, and in providing practical methods and solutions for representing states in the complex domain of biology. In addition, an important goal of our work has been to provide a layer of abstraction above an action language that can be conveniently used as the target output of NLP programs and by domain experts for authoring axioms. This layer of abstraction, also referred to as an ontology design pattern (ODP) (Gangemi and Chaudhri 2009), is similar to a process flow diagram in the sense that it captures the steps and participants of a process.

In implementing our solution, we used an off-the-shelf action representation language called $\mathcal{ALM}$ (Gelfond and Inclezan 2009; Inclezan and Gelfond 2011) and an off-the-shelf upper ontology called Component Library (CLIB)

(Barker, Porter, and Clark 2001). However, our method can be easily applied to other upper ontologies and formalisms for representing background knowledge about actions. We hope that this paper will reconcile and bring closer the ontology, NLP, and the action language communities. Such work is essential, as it would enable reusing knowledge bases developed by researchers in those communities.

We will begin this paper by introducing examples from the biology textbook that involve states. We then address each of the three questions that we posed at the outset, and show our solutions.

Example of States from a Biology Textbook

We will consider four examples of states: (1) developmental states, (2) states of systems, (3) states of molecules, and (4) states of chromosomes. We will introduce each example using relevant sentences directly from the textbook.

Developmental States

Here are some salient sentences that capture the animal developmental states:

S1 In most animal species, a small, flagellated sperm fertilizes a larger, nonmotile egg, forming a diploid zygote.

S2 The zygote then undergoes cleavage, a succession of mitotic cell divisions without cell growth between the divisions.

S3 During the development of most animals, cleavage leads to the formation of a multicellular stage called a blastula, which in many animals takes the form of a hollow ball.

S4 Following the blastula stage is the process of gastrulation, during which the layers of embryonic tissues that will develop into adult body parts are produced. The resulting developmental stage is called a gastrula.

S5 Animals eventually undergo metamorphosis, a developmental transformation that turns into a juvenile that resembles an adult but is not yet sexually mature.

We will only focus on those aspects of these sentences that pertain to actions and states. Abstractly, the animal developmental process is a complex process consisting of the following sequence of steps: Fertilization, Cleavage, Gastrulation, and Metamorphosis. These steps respectively produce a Zygote, Blastula, Gastrula, and Juvenile. Here, Zygote, Blastula, Gastrula, and Juvenile are concepts that have a flavor of states, as they refer to the evolving state of an animal as it goes through the process of development.

States of Systems

Here are some salient sentences that introduce Homeostasis and Equilibrium.

S6 Faced with environmental fluctuations, animals regulate (control) certain internal variables while allowing other internal variables to conform to (correspond to) external changes. Homeostasis is the maintenance of a steady state despite internal and external changes.

S7 The point at which the reactions offset one another exactly is called chemical equilibrium. This is a dynamic equilibrium; reactions are still going on, but with no net effect on the concentrations of reactants and products.

During the state of Homeostasis, the animal undergoes changes (e.g., its temperature may fluctuate between the regulatory limits, but as long as the temperature stays within those limits, the animal is considered to be in the state of homeostasis). Equilibrium is used to describe the state of a system of reactions. At the point of Equilibrium, the reactions are still in progress, and thus, we do not have a static state of the system.

States of Molecules

Here are example sentences that introduce states of molecules.

S8 The key to coupling exergonic and endergonic reactions is the formation of this phosphorylated intermediate, which is more reactive (less stable) than the original unphosphorylated molecule.

S9 Such methylation silences the allele, an effect consistent with evidence that heavily methylated genes are usually inactive.

Here, the molecule changes its states from being unphosphorylated to phosphorylated, and from being unmethylated to being methylated.

States of Processes

Consider the following sentences that introduce states of processes.

S10 Bacteria living deep in Earth's crust may suspend their growth for more than a century without dividing, then multiply for a few days before once again suspending growth.

S11 If all developmental activity is suspended in a seed, even when conditions appear to be suitable for its growth, the seed is said to be dormant.

S12 When the repressor is bound, transcription of the operon is blocked.

S13 Cell cycle is blocked at G1, allowing differentiation to occur.

Sentences S10 and S11 describe a suspended growth process and development process. In sentences S12 and S13, transcription and cell cycle are blocked. The usage of state concepts here is very similar to their usage in operating system processes, where processes are initiated, blocked, terminated, or suspended (Silberschatz, Galvin, and Gagne 2009).

What Counts as a State?

The definition of a state in the traditional literature is pretty straightforward: it refers to a possible state of the world (McCarthy and Hayes 1968; Reiter 1996; 2001). The actions are responsible for taking the system from one state to another (Harel 1987; Henzinger 2000).

We can apply the classical notion of states to our examples as follows. The developmental states such as Zygote, Blastula, etc. are states, and the developmental processes such Fertilization, Gastrulaiton, etc. are state transitions. Homeostasis is a state, with external stimuli and regulatory processes that are not specified in sentence S6 as transitions. Analogously, Equilibrium is a state, with forward and backward reactions being state transitions. A system of reactions can be in other states such as when the equilibrium is towards the left or the right depending on whether the rate of forward reaction is greater or less than the rate of the backward reaction, however, that detail is not specified in S7. For S8, a molecule can be in two states such as phosphorylated and unphosphorylated, with the Phosphorylation reaction being a state transition. The Methylation reaction can be modeled in an analogous manner. For sentence S10, we can define active and suspended as two states, with Activate and Suspend actions performed by the Bacteria as the transitions between the two. The sentences S11-S13 can be handled in a similar fashion.

The straightforward application of the classical definition of states to the examples considered here is ontologically neutral as it treats both entity and event states uniformly. CLIB, just like other upper ontologies such as DOLCE and SUMO, distinguishes between a Entity and an Event and defines them to be disjoint classes. We observe that concepts such as Zygote and Blastula are entities, but the concepts such as Homeostasis and Equilibrium are not entities. The impact of this distinction on knowledge engineering is that for entities, we are interested in describing their physical and spatial structure, their material constituents, different regions, etc., whereas for events, we are interested in describing their event structure, the temporal ordering of different steps, their participants, etc. Thus, to model the range of state concepts considered here, we have found it necessary to distinguish between the state of an entity and state of a process — a distinction that is not present in the traditional definition of states (Reiter 2001; Harel 1987; Henzinger 2000).

The examples for the states of molecules (e.g., phosphorylated, methylated) and the states of processes (e.g., suspended, blocked) raise the question about what state concepts should be added to the ontology. For example, should we introduce a new class in the ontology called a Phosphorylated-Molecule or just have a new Boolean property such as phosphorylated-p? Although the two solutions are logically equivalent, it is important to take into account their usage by domain experts, and also set up clear knowledge engineering guidelines that can be uniformly applied across the project.

Distinguishing between the state of an entity and the state of a process offers great convenience by providing appropriate applicable relations for describing them in a KB. Furthermore, it enables us to bring our representation closer to the way the domain experts think about the concepts in a domain, making the representation cognitively more valid and easier to understand.

There has been prior work on natural actions (Reiter 1996) and hybrid systems (Henzinger 2000) that may superficially suggest a solution to modeling actions in the biology domain. But a closer examination reveals that in hybrid systems, a process is associated with a function to define how this process changes certain properties of the world in continuous time. For instance, in the case of a falling ball, its process description would include a function that defines the height of the ball at different time points in continuous time. In many cases, the duration of the process is known. On the contrary, for the processes found in a biology textbook, such functions or durations are not specified because either they are not known or not relevant to the explanation/ understanding of the process. Therefore, the problem of modeling natural actions and hybrid systems is orthogonal to the problem of creating an ontology aware representation of states.

State of an Entity

We define the state of an entity as an entity such that its relation or property values change during an action, but it maintains its identity during those changes. We have encountered two variations of entity states in the textbook: entity states that have specific biological names, and entity states that do not have biological names. For the situation where an entity state does not have a biological name, it can be defined by assigning values to its time-varying relations, (i.e., its fluent relations) (Reiter 2001).

In the examples considered above, Zygote, Blastula, and Gastrula are examples of entity states that have a biological name. Methylated gene and phosphorylated molecule are examples of entity states that do not have biological names. A methylated gene can be represented in the KB either by creating a concept such as Methylated-Gene or by creating a fluent property methylated-p that can take a value of true or false. These two representations are logically equivalent because a Gene is a subclass of Methylated-Gene if and only if the value of methylated-p is true. Introducing fluent relationships is, however, preferable because the membership of a Gene in the class Methylated-Gene is time-dependent, and in most ontologies, the subclass relationships are expected to hold true at all times (Guarino and Welty 2002).

State of a Process

The state of a process is a distinguished sub-interval in its execution that persists for a period of time. It is distinguished in the sense that it has some features that are specifically called out in the textbook, the textbook refers to it as a state of the process, and is usually associated with a system (i.e., a set of entities). Some properties of the entities in the system or the process itself may remain constant during this state, but some properties may change.

Because the state of a process is a distinguished sub-interval in its execution, it is itself a process. A process state can have the same relationships as the steps of a process (e.g., next, prev, agent, object, etc.). In natural language, sometimes the word *phase* is used to refer to process states.

In the examples above, Homeostasis and Equilibrium are process states. For the suspended growth of bacteria, no specific biological name exists. Such process states can be handled in a manner analogous to entity states that have no biological name: by defining a property such as suspended-p, which can have a value of true or false.

This definition admits usage such as: *A cell cycle is in the state of mitosis*. Though such usage is correct in natural language, is often seen in the textbook, it does not offer any new functionality because it is redundant with a representation construct that indicates that the cell cycle is currently executing the mitosis step.

Discussion

The notion of an entity state and a process state that we have considered in this section is orthogonal to the notion of situation considered in situation calculus. Although the textbook uses the word *state* in the examples we considered, the situations of situation calculus are not an appropriate representation construct to model them.

In an alternative modeling approach using state transition diagrams, one could represent animal states such as Zygote, Blastula, and Gastrula as states, and the animal development processes such as Fertilization, Cleavage, and Gastrulation as transitions among them. Such an approach ontologically too neutral as it does not offer a direct way to capture the structural properties (e.g., blastula is a hollow ball, and that it is multi-cellular) about these concepts that can only be associated with entities.

If one uses situation calculus in this representation, one can imagine states such as Be-Gastrulated as advocated in the Component Library (Barker, Porter, and Clark 2001), but such concepts are not useful to represent any information that is discussed in the textbook.

Where to put State Concepts in a Taxonomy?

Our knowledge base `KB_Bio_101` uses the upper ontology called Component Library (CLIB) (Barker, Porter, and Clark 2001). In this section we report on, and justify, our placement of state concepts in this ontology. Our methodology can be easily expanded to other ontologies.

As entity states are entities, a natural placement for them in a taxonomy is under an appropriate subclass of Entity. A Blastula could not be made a subclass of Animal – a subclass of Entity in CLIB – because it does not have the fully developed structure of an Animal. A superclass of Animal in CLIB is Living − Entity, but whether Blastula should be considered a Living − Entity is unclear, therefore, we place it as a subclass of Entity.

In CLIB, the class Event has two subclasses: Action and State. The class State has a further subclass of Process-State. The concepts such as Homeostasis and Equilibrium will be placed as subclasses of Process-State. The class State could have other state concepts such as Be-Attached as its subclasses.

How to Use State Concepts in Representation?

In this section, we consider in detail one of the state concepts – *developmental state* – and illustrate its formalization. Due to space considerations, we do not address other types of state concepts here.

We first show the formalization of developmental states using the process flow chart ODP, and then consider the background axioms associated with it. The background axioms were written in a language called $\mathcal{ALM}$ (Gelfond and Inclezan 2009). $\mathcal{ALM}$ is an action language (Gelfond and Lifschitz 1993), i.e., a declarative language dedicated to the representation of actions, their effects, and preconditions. Theories of $\mathcal{ALM}$ are translated into logic programs under the answer set semantics (ASP) (Gelfond and Lifschitz 1991) that can be used in solving non-trivial reasoning tasks. For readability purposes, instead of showing axioms written in $\mathcal{ALM}$, we include here their translation into ASP.

Our methodology of dividing the representation task into a process flow chart and background axioms corresponds to the division between two types of encoders of `KB_Bio_101`: domain experts in biology and knowledge engineers. The domain specific information is entered by domain experts using a graphical interface; thus domain experts do not need to have extensive training in knowledge representation. The background axioms are written once and for all by knowledge engineers who do not need to learn the intricacies of the domain specific knowledge. A simple algorithm for translating the process flow chart into ASP axioms allows us to combine the two components and use them in reasoning.

Process Flow Chart Representation of Developmental States

The graphical representation of the ODP for the process of animal development can be seen in Figure 1. The ODP has a straightforward translation into logic programming notation, which consists of: (a) facts specifying that animal, zygote, blastula, gastrula, and juvenile are subclasses of the class entity; (b) facts saying that animal_development is a subclass of event, and that fertilization, cleavage, gastrulation, and metamorphosis are actions; (c) axioms instantiating entities and actions related to the process of animal development; and (d) axioms encoding the relations appearing in the graph in Figure 1.

In equations (1) and (2) below, we give a few examples of axioms of type (c) and (d), respectively. Note that, in ASP, identifiers starting with an uppercase letter denote variables; identifiers starting with a lowercase letter denote constant symbols. Let us explain the meaning of the key symbols and relations used in (1) and (2). Functions a, f, c, and z with arity 1 are used to reify the animal, fertilization phase, cleavage phase, and zygote state, respectively, associated with a specific animal development instance (e.g., $a(x)$ is the animal that undergoes the animal development process denoted by x). The relation $\mathrm{subevent}(x, y)$ says that y is a subevent of process x; relation $\mathrm{first\_event}(x, y)$ denotes that y is the first step of the process x; $\mathrm{next\_event}(x, y)$ denotes that y happens after x; $\mathrm{object}(x, y)$ says that x is the object of process y; $\mathrm{raw\_material}(x, y)$ means that y is an input to process x; $\mathrm{result}(x, y)$ means that y is the result of process x. Using the relations considered so far, we can state the steps involved in the animal development process, their relative ordering, and input/output to each step. The last relation, $\mathrm{has\_state}(x, y)$, requires some discussion. Recall that, during the process of animal development, the animal retains its identity (i.e., as it goes through the various stages of development such as Zygote, Blastula, etc., it is still the same animal). We capture this by using the $\mathrm{has\_state}$ relationship: $\mathrm{has\_state}(x, y)$ indicates that x can *potentially* be in state y at a given moment in time.

$$
\begin{array}{l}
is\_a(a(X), animal) \leftarrow is\_a(X, animal\_development) \\
is\_a(f(X), fertilization) \leftarrow is\_a(X, animal\_development) \\
is\_a(c(X), cleavage) \leftarrow is\_a(X, animal\_development) \\
is\_a(z(X), zygote) \leftarrow is\_a(X, animal\_development)
\end{array} \quad (1)
$$

$$
\begin{array}{l}
subevent(X, f(X)) \leftarrow is\_a(X, animal\_development) \\
first\_event(X, f(X)) \leftarrow is\_a(X, animal\_development) \\
next\_event(f(X), c(X)) \leftarrow is\_a(X, animal\_development) \\
object(X, a(X)) \leftarrow is\_a(X, animal\_development) \\
raw\_material(X, a(X)) \leftarrow is\_a(X, animal\_development) \\
raw\_material(f(X), a(X)) \leftarrow is\_a(X, animal\_development) \\
result(f(X), z(X)) \leftarrow is\_a(X, animal\_development) \\
has\_state(a(X), z(X)) \leftarrow is\_a(X, animal\_development)
\end{array} \quad (2)
$$

Background Axioms

The background knowledge can be expressed using five categories of axioms: (a) state constraints (also known as *static causal laws*); (b) successor state axioms (also called *dynamic causal laws*); (c) impossibility conditions; (d) closed world assumptions; and (e) inertia axioms. We also include a theory of intentions that consists of a set of axioms that define general purpose properties of actions.

We start with state constraints. We define relation $\mathrm{instance\_of}$ as the transitive closure of the basic relation $\mathrm{is\_a}$ that is used in our encoding via the following two axioms:

$$
\begin{array}{lcl}
instance\_of(X, S) & \leftarrow & is\_a(X, S) \\
instance\_of(X, S) & \leftarrow & is\_a(X, S1), \\
 & & subclass(S1, S)
\end{array} \quad (3)
$$

Another state constraint extends relation object from processes to steps of processes (i.e., subevents):

$$
\begin{array}{lcl}
object(X, A) & \leftarrow & instance\_of(X, action), \\
 & & instance\_of(A, entity), \\
 & & instance\_of(Y, event), \\
 & & subevent(Y, X), \\
 & & object(Y, A)
\end{array} \quad (4)
$$

We view processes like animal development as discrete processes; therefore, we can associate an integer step identifier with the occurrence of each step of the process. In what follows, we will use the variable I for discrete time steps. During the process, the state of the animal changes, but the collection of rules 1-2 does not directly indicate this. The state change can be understood using the following successor state axiom that uses a ternary version of the $\mathrm{has\_state}$ relationship, and a new relation occurs that denotes whether a step actually happens.

$$
\begin{array}{lcl}
has\_state(A, C, I+1) & \leftarrow & occurs(X, I), \\
 & & object(X, A), \\
 & & result(X, C)
\end{array} \quad (5)
$$

The following impossibility conditions specify when an action cannot happen: if the current state does not match the $\mathrm{raw\_material}$ state, or if the animal is already in the result state:

$$
\begin{array}{lcl}
\neg occurs(X, I) & \leftarrow & object(X, A), \\
 & & raw\_material(X, B), \\
 & & \neg has\_state(A, B, I)
\end{array} \quad (6)
$$

$$
\begin{array}{lcl}
\neg occurs(X, I) & \leftarrow & object(X, A), \\
 & & result(X, C), \\
 & & has\_state(A, C, I)
\end{array} \quad (7)
$$

Axiom 5 makes clear that, as the process executes, the state of the entity changes from one step to the next. Here, the symbol $\neg$ denotes classical negation. Axiom 6 says that, if the entity is not in the state that is necessary for a step to occur, that step does not occur. Finally, axiom 7 says that, if the entity is already in the state that results from a step, then that step does not occur.

Next, we consider closed world assumptions for each of the relations. We show here only the closed world assumption for relation result. Similar axioms exist for the relations $\mathrm{raw\_material}$, object, $\mathrm{subevent}$, $\mathrm{first\_event}$, and $\mathrm{next\_event}$. In rule 8 below, we say that, if we have no reason to believe that an entity A is the result of an action X, we can conclude that it is indeed not a result of that action.

$$
\begin{array}{lcl}
\neg result(X, A) & \leftarrow & instance\_of(X, action), \\
 & & instance\_of(A, entity), \\
 & & \text{not } result(X, A)
\end{array} \quad (8)
$$

We consider the inertia axioms for the $\mathrm{has\_state}$ relationship, the only relation that may change in time.

$$
\begin{array}{l}
has\_state(A, B, I+1) \leftarrow \\
\quad instance\_of(A, entity), \\
\quad instance\_of(B, entity), \\
\quad has\_state(A, B, I), \\
\quad \text{not } \neg has\_state(A, B, I+1) \\
\neg has\_state(A, B, I+1) \leftarrow \\
\quad instance\_of(A, entity), \\
\quad instance\_of(B, entity), \\
\quad \neg has\_state(A, B, I), \\
\quad \text{not } has\_state(A, B, I+1)
\end{array} \quad (9)
$$

The inertia axioms in 9 capture the fact that, if an entity is in a particular state (or not in a particular state), and we cannot prove that its state changed at the next step, then it continues to be (or not be) in that state.

Finally, we introduce a theory that captures the unfolding of processes in the event of unexpected interruptions, the *theory of intentions* (Baral and Gelfond 2005). We assume that biological entities have the intrinsic "intention" to go through all the steps in their development. The main tenets of the theory of intentions are: *"Normally intended actions are executed the moment such execution becomes possible"* (non-procrastination) and *"Unfulfilled intentions persist"* (persistence).

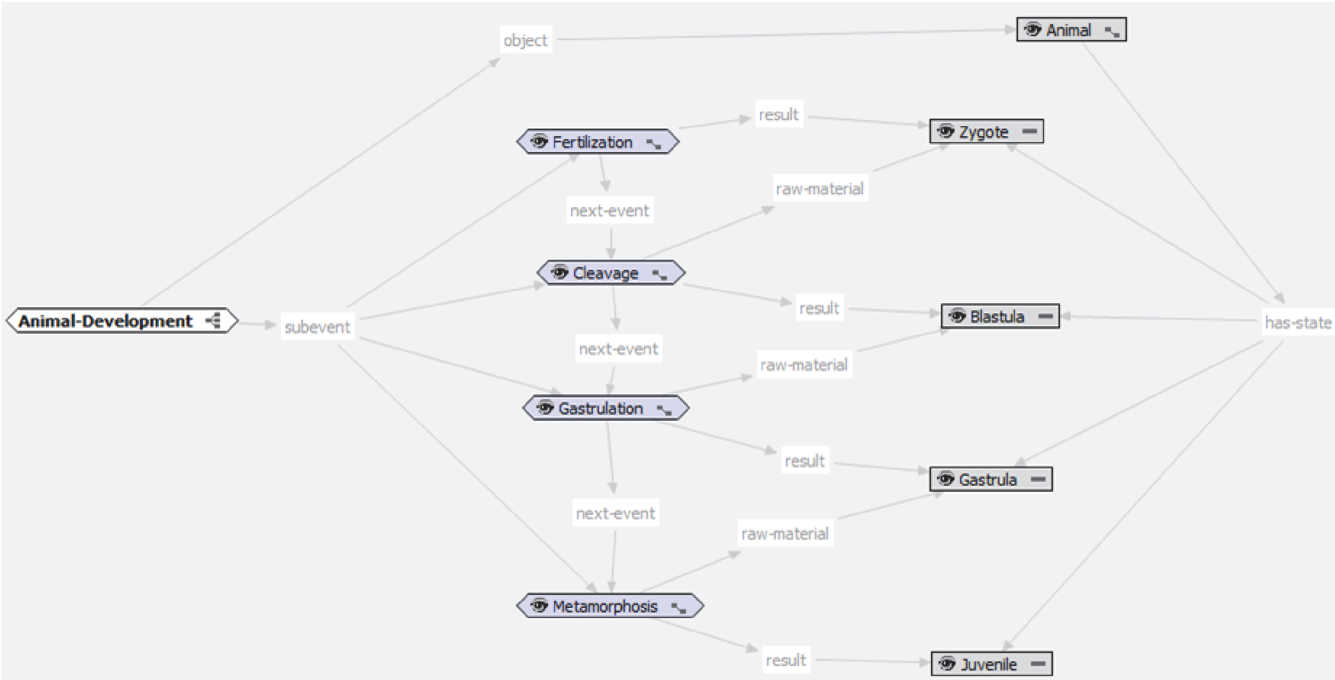

Figure 1: Graphical visualization of the process flow diagram for animal development.

Using the Axioms for Answering Questions

The background axioms 3-9 provide precise semantics in terms of *transition diagrams*, as defined by the action languages. Using established methodologies from logic programming, our theory of developmental processes can be applied to answering complex questions about process interruption (Inclezan and Gelfond 2011) as we show next.

Example 1 *A sheep embryo in a research environment has gone through fertilization, cleavage, and gastrulation. A researcher now introduces a chemical that prevents metamorphosis from occurring. In what stage will the embryo be?*

We start by representing the facts specified in the text of the question. First, we introduce a name for the process of *sheep* animal development:

$is\_a(sheep\_dev, animal\_development)$

We know that three of the process steps occurred at consecutive time steps; 0 indicates the initial time step in our story.

$occurs(f(sheep\_dev), 0)$
$occurs(c(sheep\_dev), 1)$
$occurs(g(sheep\_dev), 2)$

We know that the chemical introduced by the researcher prevents the last process step from occurring at any time step I:

$\neg occurs(m(sheep\_dev), I)$

By putting together the encoding of the ODP chart, our background axioms, the encoding of the information given in the question, and the theory of intentions, we obtain a logic program. Models (i.e., answer sets) of this program can be computed using off-the-shelf solvers (e.g., CLASP, DLV). The program will have only one answer set, which will contain:

$has\_state(a(sheep\_dev), zygote, 1)$
$has\_state(a(sheep\_dev), blastula, 2)$
$has\_state(a(sheep\_dev), gastrula, 3)$
$has\_state(a(sheep\_dev), gastrula, 4)$
$has\_state(a(sheep\_dev), gastrula, 5) \ldots$

The $\mathrm{juvenile}$ stage is never reached, and the sheep embryo remains in the $\mathrm{gastrula}$ stage as expected. This result is in fact the answer to the question in Example 1.

Summary and Conclusions

Our work extends the prior work on modeling natural actions and hybrid systems by developing an ontology aware representation of states. Specifically, we distinguish between entity states and process states, a distinction that is absent in the prior literature, and give definitions that can be used in practical knowledge-engineering projects. We also define a layer of abstraction, called a process flow diagram, that is more abstract than an action representation language and is better suited for knowledge authoring by domain experts and as a target for the output of NLP programs. We show that a combination of a process flow chart representation with background axioms leads to a powerful computational tool applicable for simulating processes and answering process-interruption reasoning questions. We hope that this work is a step towards bridging the gap between linguistic, ontological, and action representations.

Acknowledgments

This work work has been funded by Vulcan Inc. and SRI International. We thank Nikhil Dinesh and Michael Gelfond for numerous discussions and feedback as we developed the ideas presented in this paper.

References

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, 86–90.

Banik, E.; Kow, E.; and Chaudhri, V. K. 2013. User-controlled, robust natural language generation from an evolving knowledge base. In *ENLG 2013 : 14th European Workshop on Natural Language Generation*.

Baral, C., and Gelfond, M. 2005. Reasoning about intended actions. In *AAAI-05: Proceedings of the 20th National Conference on Artificial Intelligence*, 689–694. AAAI Press.

Baral, C. 2003. *Knowledge representation, reasoning, and declarative problem solving*. Cambridge University Press.

Barker, K.; Porter, B.; and Clark, P. 2001. A library of generic concepts for composing knowledge bases. In *First 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. Query answering in object oriented knowledge bases in logic programming. In *Workshop on ASP and Other Computing Paradigms*.

Chaudhri, V. K.; Elenius, D.; Hinojoza, S.; and Wessel, M. 2014. KB Bio 101: Content and challenges. In *Proceedings of International Conference on Formal Ontologies in Information Systems*.

Gangemi, A., and Chaudhri, V. K. 2009. Representing the component library into ontology design patterns. *2009 ISWC Workshop on Ontology Design Patterns*.

Gangemi, A.; Guarino, N.; Masolo, C.; and Oltramari, A. 2003. Sweetening WordNet with DOLCE. *AI magazine* 24(3):13.

Gelfond, M., and Inclezan, D. 2009. Yet Another Modular Action Language. In *Proceedings of SEA-09*, 64–78. University of Bath Opus: Online Publications Store.

Gelfond, M., and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9(3/4):365–386.

Gelfond, M., and Lifschitz, V. 1993. Representing action and change by logic programs. *The Journal of Logic Programming* 17(2):301–321.

Guarino, N., and Welty, C. 2002. Evaluating ontological decisions with OntoClean. *Communications of the ACM* 45(2):61–65.

Harel, D. 1987. Statecharts: A visual formalism for complex systems. *Science of computer programming* 8(3):231–274.

Henzinger, T. A. 2000. *The theory of hybrid automata*. Springer.

Inclezan, D., and Gelfond, M. 2011. Representing Biological Processes in Modular Action Language ALM. In *Proceedings of the 2011 AAAI Spring Symposium on Formalizing Commonsense*, 49–55. AAAI Press.

Kipper, K.; Korhonen, A.; Ryant, N.; and Palmer, M. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation* 42(1):21–40.

Lenat, D. B. 1995. CYC: A large scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

McCarthy, J., and Hayes, P. 1968. *Some philosophical problems from the standpoint of artificial intelligence*. Stanford University.

Niles, I., and Pease, A. 2001. Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems-Volume 2001*, 2–9. ACM.

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.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *Proceedings of the International conference on Knowledge Representation and Reasoning*, 2–13.

Reiter, R. 2001. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press.

Silberschatz, A.; Galvin, P. B.; and Gagne, G. 2009. *Operating system concepts*. J. Wiley & Sons.