WordsEye: An Automatic Text-to-Scene Conversion System

Bob Coyne Richard Sproat

AT&T Labs — Research*

Abstract

Natural language is an easy and effective medium for describing visual ideas and mental images. Thus, we foresee the emergence of language-based 3D scene generation systems to let ordinary users quickly create 3D scenes without having to learn special software, acquire artistic skills, or even touch a desktop window-oriented interface. WordsEye is such a system for automatically converting text into representative 3D scenes. WordsEye relies on a large database of 3D models and poses to depict entities and actions. Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process. We describe the linguistic analysis and depiction techniques used by WordsEye along with some general strategies by which more abstract concepts are made depictable.

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial Realities H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input Devices and Strategies I.2.7 [Artificial Intelligence]: Natural Language Processing—Language Parsing and Understanding I.3.6 [Graphics]: Methodologies—Interaction Techniques I.3.8 [Graphics]: Applications

Keywords: Applications, HCI (Human-Computer Interface), Multimedia,Text-to-Scene Conversion, Scene Generation

1 Introduction

Creating 3D graphics is a difficult and time-consuming process. The user must learn a complex software package, traverse pages of menus, change tools, tweak parameters, save files, and so on. And then there's the task of actually creating the artwork. We see the need for a new paradigm in which the creation of 3D graphics is both effortless and immediate. It should be possible to describe 3D scenes directly through language, without going through the bottleneck of menu-based interfaces. Creating a 3D scene would then be as easy as dashing off an instant message.

Natural language input has been investigated in a number of 3D graphics systems including an early system by [2] and the oft-cited Put system [8]; the Put system shared our goal of making graphics creation easier, but was limited to spatial arrangements of existing objects. Also, input was restricted to an artificial subset of English consisting of expressions of the form *Put (X P Y)*$^+$, where *X* and *Y* are objects, and *P* is a spatial preposition. The system did allow for fairly sophisticated interpretation of spatial relations so that *on* in *on the table* and *on the wall* would both be appropriately interpreted. More recently, there has been work at the University of Pennsylvania's Center for Human Modeling and Simulation [3, 4], where language is used to control animated characters in a closed virtual environment. In previous systems the referenced objects and actions are typically limited to what is available and applicable in the pre-existing environment. These systems therefore have a natural affinity to the SHRDLU system [23] which, although it did not have a graphics component, did use natural language to interact with a "robot" living in a closed virtual world.

* {coyne,rws}@research.att.com

Figure 1: *John uses the crossbow. He rides the horse by the store. The store is under the large willow. The small allosaurus is in front of the horse. The dinosaur faces John. A gigantic teacup is in front of the store. The dinosaur is in front of the horse. The gigantic mushroom is in the teacup. The castle is to the right of the store.*

The goal of WordsEye, in contrast, is to provide a blank slate where the user can literally paint a picture with words, where the description may consist not only of spatial relations, but also actions performed by objects in the scene. The text can include a wide range of input. We have also deliberately chosen to address the generation of static scenes rather than the control or generation of animation. This affords us the opportunity to focus on the key issues of semantics and graphical representation without having to address all the problems inherent in automatically generating convincing animation. The expressive power of natural language enables quite complex scenes to be generated with a level of spontaneity and fun unachievable by other methods (see Figure 1); there is a certain magic in seeing one's words turned into pictures.

WordsEye works as follows. An input text is entered, the sentences are tagged and parsed, the output of the parser is then converted to a *dependency structure*, and this dependency structure is then semantically interpreted and converted into a *semantic representation*. *Depiction rules* are used to convert the semantic representation to a set of low-level *depictors* representing 3D objects, poses, spatial relations, color attributes, etc; note that a *pose* can be loosely defined as a character in a configuration suggestive of a particular action. Transduction rules are applied to resolve conflicts and add implicit constraints. The resulting depictors are then used to manipulate the 3D objects that constitute the final, renderable 3D scene. For instance, for a short text such as: *John said that the cat was on the table. The animal was next to a bowl of apples*, WordsEye would construct a picture of a human character with a cartoon speech bubble coming out of its mouth. In that speech bubble would be a picture of a cat on a table with a bowl of apples next to it.[1]

[1] With the exception of the initial tagging and parsing portion of the linguistic analysis, WordsEye is implemented in Common Lisp, runs within

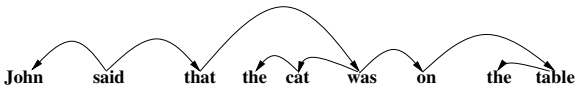

Figure 2: Dependency structure for *John said that the cat was on the table.*.

Since linguistic descriptions tend to be at a high level of abstraction, there will be a certain amount of unpredictability in the graphical result. This same tradeoff is seen in computational models of behavior [21, 12] and natural phenomena. We also acknowledge up front that it is infeasible to fully capture the semantic content of language in graphics. But we do believe that a large number of interesting 3D scenes can be described and generated directly through language, and likewise that a wide variety of text can be effectively depicted.

In the remainder of this paper, we describe each of the components of WordsEye, starting with an overview of the linguistic analysis techniques used.

2 Linguistic Analysis

The text is initially tagged and parsed using a part-of speech-tagger [7] and a statistical parser [9]. The output of this process is a parse tree that represents the structure of the sentence. Note that since the parser is statistical, it will attempt to resolve ambiguities, such as prepositional phrase attachments, according to the statistics of the corpus on which it was trained (the Penn Treebank [19]). The parse tree is then converted into a *dependency representation* (see [16], inter alia) which is simply a list of the words in the sentence, showing the words that they are dependent on (the *heads*) and the words that are dependent on them (the *dependents*). Figure 2 shows an example dependency structure, with arrows pointing from heads to their dependents. The reason for performing this conversion from parse tree to dependency structure is that the dependency representation is more convenient for semantic analysis. For example, if we wish to depict *the large naughty black cat* we might actually have no way of depicting *naughty*, but we still would like to depict *large* and *black*. To do this we need merely to look at *cat*'s dependents for depictable adjectives, which is in general simpler than hunting for depictable modifiers in a tree structure headed by *cat*.

The next phase of the analysis involves converting the dependency structure into a semantic representation. The semantic representation is a description of the entities to be depicted in the scene, and the relationships between them. The semantic representation for the sentence *John said that the cat is on the table* is given in Figure 3. The semantic representation is a list of semantic representation fragments, each fragment corresponding to a particular node of the dependency structure. Consider "node1", which is the semantic representation fragment for the action *say*, deriving from the node *say* in the dependency structure. The subject is "node2" (corresponding to *John*), and the direct object is the collection of "node5", "node4" and "node7", corresponding to nodes associated with the subordinate clause *that the cat was on the table*. Each of these nodes in turn correspond to particular nodes in the dependency structure, and will eventually in turn be depicted by a given 3D object: so *John* will be depicted (in the current system) by a humanoid figure we call "Mr. Happy", and *table* will be depicted by one of a set of available 3D table objects.[2]

the Mirai^{TM} 3D animation system from IZware, and uses 3D models from Viewpoint.

[2]An individual semantic representation fragment as currently used in WordsEye may seem relatively simple when compared, say, with the PAR representation of [3]. But bear in mind that an entire semantic representation for a whole scene can be a quite complex object, showing relations between many different individual fragments; further semantic information is expressed in the depiction rules described below. Also note that part of the complexity of PAR is due to the fact that that system is geared towards generating animation rather than static scenes.

```
(("node2" (:ENTITY :3D-OBJECTS ("mr_happy")
           :LEXICAL-SOURCE "John" :SOURCE SELF))
 ("node1" (:ACTION "say" :SUBJECT "node2"
           :DIRECT-OBJECT  ("node5" "node4" "node7")...))
 ("node5" (:ENTITY :3D-OBJECTS  ("cat-vp2842")))
 ("node4" (:STATIVE-RELATION "on" :FIGURE "node5"
           :GROUND "node7"))
 ("node7" (:ENTITY :3D-OBJECTS
           ("table-vp14364" "nightstand-vp21374"
            "table-vp4098" "pool_table-vp8359" ...))))
```

Figure 3: Semantic representation for *John said that the cat was on the table.*

Semantic representation fragments are derived from the dependency structure by semantic interpretation frames. The appropriate semantic interpretation frames are found by table lookup, given the word in question. These frames differ depending upon what kind of thing the word denotes. For nouns such as *cat* or *table*, WordsEye uses WordNet [10], which provides various kinds of semantic relations between words, the particular information of interest here being the *hypernym* and *hyponym* relations. The 3D objects are keyed into the WordNet database so that a particular model of a cat, for example, can be referenced as *cat*, or *feline* or *mammal*, etc. Personal names such as *John* or *Mary* are mapped appropriately to male or female humanoid figures. Spatial prepositions such as *on* are handled by *semantic functions* that look at the left and right dependents of the preposition and construct a semantic representation fragment depending upon their properties. Note that there has been a substantial amount of previous work into the semantics of spatial prepositions; see, inter alia, [5, 14, 15] and the collections in [11, 20]; there has also been a great deal of interesting cross-linguistic work, as exemplified by [22]. There have been only a small number of implementations of these ideas however; one sophisticated instance is [24]. One important conclusion of much of this research is that the interpretation of spatial relations is often quite object-dependent, and relates as much to the *function* of the object as its geometric properties, a point that ties in well with our use of *spatial tags*, introduced below in Section 3.1.

Finally, most verbs are handled by semantic frames, which are informed by recent work on verbal semantics, including [18]. The semantic entry for *say* is shown in Figure 4. This semantic entry contains a set of verb frames, each of which defines the *argument structure* of one "sense" of the verb *say*. For example, the first verb frame, named the SAY-BELIEVE-THAT-S-FRAME, has as required arguments a subject and a THAT-S-OBJECT, or in other words an expression such as *that the cat is on the table*. Optional arguments include *action location* (e.g., *John said* **in the bathroom** *that the cat was on the table*) and *action time* (e.g., *John said* **yesterday** *that the cat was on the table*.) Each of these argument specifications causes a function to be invoked to check the dependencies of the verb for a dependent with a given property, and assigns such a dependent to a particular slot in the semantic representation fragment. WordsEye currently has semantic entries for about 1300 English nouns (corresponding to the 1300 objects described in Section 3.1), and about 2300 verbs, in addition to a few depictable adjectives, and most prepositions. The vocabulary is, however, readily extensible and is limited only by what we are able to depict.

In addition to semantically interpreting words that denote par-

```
(SEMANTICS :GENUS say
 :VERB-FRAMES
   ((VERB-FRAME
     :NAME SAY-BELIEVE-THAT-S-FRAME
     :REQUIRED (SUBJECT THAT-S-OBJECT)
     :OPTIONAL (ACTIONLOCATION ACTIONTIME))
    (VERB-FRAME
     :NAME SAY-BELIEVE-S-FRAME
     :REQUIRED (SUBJECT S-OBJECT)
     :OPTIONAL (ACTIONLOCATION ACTIONTIME)) ...))
```

Figure 4: Semantic entry for *say*.

ticular entities, actions or relations, WordsEye also interprets anaphoric or coreferring expressions. Simple pronominals like *he* or *she*, are interpreted by searching through the context to find an appropriate coreferent (where appropriate includes matching on number and gender features). Nouns can also corefer, as in the example: *John said that the cat was on the table. The animal was next to a bowl of apples.* While it is not strictly required that *the animal* denote the cat mentioned in the previous sentence, the coherence of the discourse depends upon the reader or listener making that connection. WordsEye currently handles such associations by noting that in the WordNet hierarchy, the denotations of *cat* are a subset of the denotations of *animal*, and "guessing" that the noun phrase might corefer with the previously mentioned *cat* (see, e.g., [13]).

Before closing the discussion of the natural language component of WordsEye, it is worth noting two points. First of all, WordsEye of necessity performs a fairly deep semantic analysis of the input text. This contrasts with what counts for "understanding" in much recent work on, for example, Message Understanding (MUC) (see, e.g., [1]); in the MUC context one is typically trying to answer a small number of questions about the text (e.g., who left which company to head up which other company), and so most approaches to understanding in this context eschew a complete semantic analysis of sentences. In WordsEye we do not have this luxury since the number of "questions" to be answered in order to construct a scene is in principle unbounded. Second, WordsEye owes an intellectual debt to work in Cognitive Grammar [17]. While WordsEye is not strictly speaking an implementation of this theory, Cognitive Grammar's model of semantics is like WordsEye's in that it constructs meanings of utterances out of graphical components that combine in ways that are similar to WordsEye's depiction phase, which we now describe.

3 Depictors

All scenes are ultimately defined in terms of a set of low-level graphical specifications which we call *depictors*. Depictors exist to control 3D object visibility, size, position, orientation, surface color and transparency. Depictors are also used to specify poses of human characters, control Inverse Kinematics (IK), and modify vertex displacements for facial expressions or other aspects of the objects. In this section we examine depictors in more detail.

3.1 Object Database

The basic elements of any 3D scene are the objects themselves. WordsEye currently utilizes approximately 2000 3D polygonal objects, with another 10,000 in the process of being integrated into the system. The majority of the database is licensed from Viewpoint Digital and includes models for animal and human characters as well as buildings, vehicles, household items, plants, etc. Since WordsEye is extensible, users can add their own models to the database.

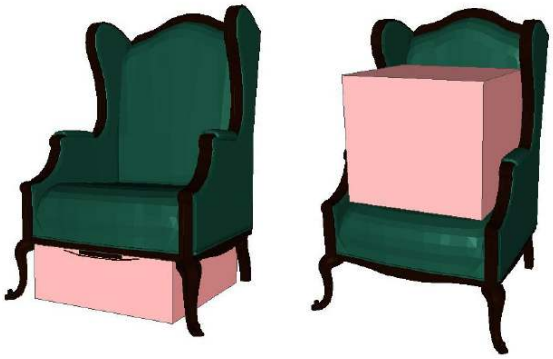

Figure 5: Spatial tag for "canopy area", indicated by the box under the lefthand chair; and "top surface", indicated by the box on the righthand chair.

In addition to the raw 3D data, WordsEye associates additional information with each 3D model.

Skeletons: Objects can contain skeletal control structures. These are used in human and animal characters to define poses representing different actions.

Shape displacements: Some objects, like human faces, can change shape (e.g., smiling, eyes closed, frowning, etc.). Shape displacements are associated with the object and used to depict emotions or other states of the object.

Parts: These are named collections of polygonal faces representing significant areas of the surface. For example, the headlights, roof, and windshield of a car would be in different parts.

Color parts: These are the set of parts to be colored when the object is specified by the text as having a particular color. For example, in *the blue flower*, the petals of the flower should be colored blue, not the stem. If no color parts are specified, the largest part is colored.

Opacity parts: These are parts that get a default transparency (e.g., the glass part of a framed window).

Default size: All objects are given a default size, expressed in feet.

Functional properties: These properties are used in the depiction process to determine how an object can be used. For example, cars, bicycles, trucks, and motorcycles are all road vehicles. Then, while depicting the verb *ride*, we select among these to choose a vehicle in the sentence *John rides to the store*.

Spatial tags: In order to depict spatial relations, the shape of the objects in question must be known. We do this by associating spatial tags with all objects in the database. For example the interior area of an ashtray functions as a *cup* to contain whatever is put in it. The area is marked with a tag (a simple space-filling 3D object) representing the cupped interior. This is used when depicting spatial relations such as *in* and sometimes *on*. Some examples of spatial tags are: *canopy area* and *top surface* (Figure 5); and *base* and *cup* (Figure 6). Other spatial tags not shown here are *wall*, *cap*, *enclosed-area*, *ridge*, *peak*.

3.2 Spatial Relations

Spatial relations define the basic layout of scenes. They include relative positions as in *John is next to Mary*, distances as in *The chair is 5 feet behind the table*, and orientations as in *John is facing the abyss*. And, as discussed later, spatial relations are frequently an implicit part of actions and compound objects.

Spatial relations are often denoted by prepositions like *on*, *under*, *beyond*, etc. The exact placement of objects, in depictions of spa-

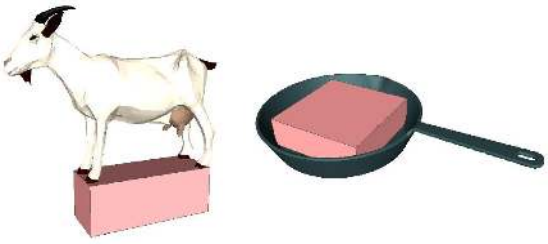

Figure 6: Spatial tags for “base” and “cup”.

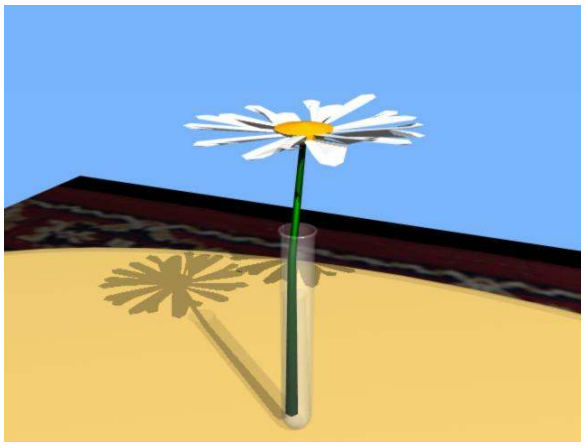

Figure 7: *The daisy is in the test tube.*

tial relations, depends on the shapes and surfaces of those objects. Additionally, terms like *in* and *under* can have different possible spatial intepretations depending on the objects in question. For example, *The cat is under the table* and *The rug is under the table* denote different spatial areas. Some examples of spatial relations are described below.

For *The bird is on the cat*, we find a *top surface* tag for the cat (on its back) and a *base* tag for the bird (under its feet). We then reposition the bird so that its feet are on the cat’s back.

For *The daisy is in the test tube*, we find the *cup* tag for the test tube and the *stem* tag for the daisy and put the daisy’s stem into the test tube’s cupped opening. See Figure 7. Spatial tags for stems are applied to any object with a long, thin base leading to a thicker or wider top area. Some objects with stems are stop signs, umbrellas, palm trees and street lamps.

For *The elephant is under the chair*, we look for a *canopy* tag for the chair (the area under the seat of the chair between the legs) and put the elephant there. This might involve resizing the elephant to make it fit. However, as noted earlier, *under* can also be interpreted so that the chair is put on the elephant’s back. Depending on the size and shape of the objects in question, one interpretation or another will be chosen. In general, we try to choose an interpretation that avoids resizing. However, we note that gross changes of scale are extremely common in advertising and often highlight the significance or functional role of the objects in question.

These examples are not meant to be an exhaustive list, but rather illustrate the manner in which we use object tags to depict spatial relations. A rendered example of a spatial relation using the *top surface* and *enclosure* spatial tags is shown in Figure 8.

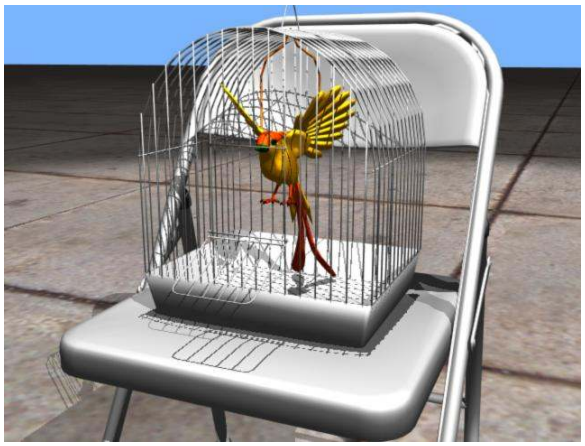

Figure 8: *The bird is in the bird cage. The bird cage is on the chair.*

Figure 9: Usage pose for a 10-speed.

3.3 Poses and Grips

Most actions are depicted in WordsEye via the use of predefined poses, where a pose can be loosely defined as a character in a configuration suggestive of a particular action.

Standalone poses consist of a character in a particular body position. Examples of this are waving, running, bowing, or kneeling.

Specialized usage poses involve a character using a specific instrument or vehicle. Some examples are swinging a baseball bat, shooting a rifle, and riding a bicycle. For a bicycle, a human character will be seated on a bicycle with its feet on the pedals and hands on the handlebars. In these, each pose is tightly associated with a particular manipulated object; see Figure 9 for an example.

Generic usage poses involve a character interacting with a generic stand-in object. The stand-in objects are represented by simple markers like spheres and cubes. We use these in cases where another object can convincingly be substituted for the stand-in. For example, in the *throw small object* pose (Figure 10, left panel), the ball is represented by a generic sphere. If the input sentence is *John threw the watermelon*, the watermelon will be substituted for the sphere in the same relative position with respect to the hand. The new object can be substituted as is or, alternatively, resized to match the size of the stand-in sphere. The positional and sizing constraints are associated with the stand-in objects and are stored in the pose.

Grip poses involve a character holding a specific object in a cer-

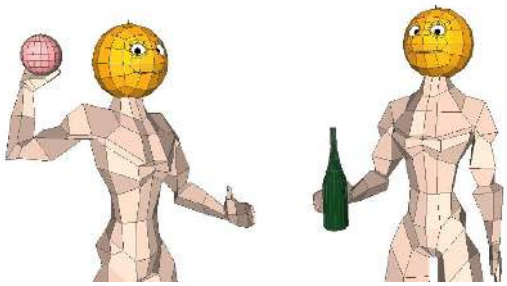

Figure 10: "Throw small object" pose and "hold wine bottle" grip.

Figure 11: *John rides the bicycle. John plays the trumpet.*

tain way. Some objects can be used in a variety of ways while being held in the same grip. For instance, if we have a grip for holding a wine bottle (Figure 10, right panel), this grip can be used in various poses, such as pouring wine, giving the bottle to someone else, putting the bottle on a surface, and so forth. This technique allows us to avoid a combinatorial explosion in the number of poses for specific objects. We do not want a separate *pour*, *give*, and *put* pose for every object in our database. We avoid this by having a small number of grips for each object and then selecting the grip appropriate for the more generic action pose. To do this, we first put and attach the object in the hand before going to the action pose. This is facilitated by classifying objects and poses into categories representing their shape. For example, the poses *swing long object* and *hold long object* might be applied to a sword in a *hold sword* grip.

Bodywear poses involve a character wearing articles of clothing like hats, gloves, shoes, etc. These are used to attach the object to the appropriate body part and are later combined with other poses and body positions.

Another strategy we adopt is to combine upper and lower body poses. Some poses require the whole body to be positioned, while for others only the upper or lower body needs to be positioned. We use the simple procedure of associating an active body part for each pose, and then moving only those bones that are necessary when more than one pose is applied to the same character. For example, see Figure 11 which shows a character riding a bicycle (lower) while playing the trumpet (upper).

3.4 Inverse Kinematics

Poses are effective for putting a character into a stance that suggests a particular action. But for a scene to look realistic and convincing, the character must sometimes interact directly with the environment. We use IK to do this [25]. So, for example, in pointing, it is not enough just to put the character into a pointing pose since the object pointed at can be anywhere in the environment. Instead, the hand must be moved with IK to point in the desired direction.

We also use IK to modify existing poses. For example, the *push large object* pose consists of the character leaning toward an object with legs in stride and arms outstretched. Consider, however, pushing various objects such as a lawnmower, a car, or a baby carriage. Since the different objects have handles and surfaces at different heights, no single body pose can work for them all. The hands must be located at the correct position on the object. To do this, the character is first put behind the object in the *push large object* pose. Then the hands are moved using IK to the handle or vertical surface of the object. Note that this technique relies on object tags for *handle* or *vertical surface* in order to determine the target position for the IK; see Figure 12, and Figure 13 for a rendered example that uses IK to move the hands to the handle of a lawnmower.

Figure 12: Spatial tag for "push handle" of baby carriage, indicated by the box around the handle.

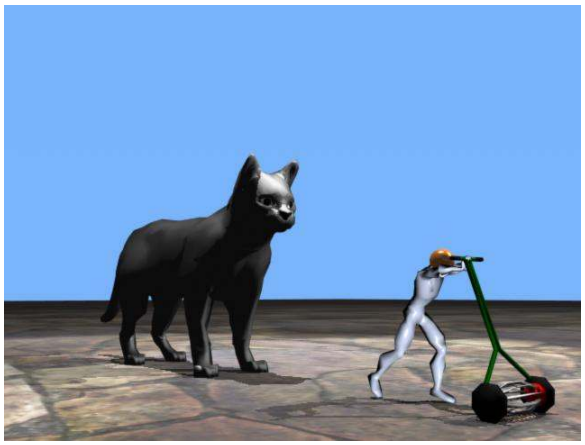

Figure 13: *The lawn mower is 5 feet tall. John pushes the lawn mower. The cat is 5 feet behind John. The cat is 10 feet tall.*

3.5 Attributes

WordsEye currently handles attributes for size, color, transparency and shape. Color and transparency are applied to the object as surface attributes. They are applied to the dominant part (as defined in the object database) of the object unless otherwise specified. The shape of the object can be modified using shape displacements in the Mirai animation system. These are predefined states of vertex positions associated with the 3D model that can be additively combined. For example, in a human face, there can be separate displacements for a smile and a wink. The various displacements can

be combined independently. We currently use this in WordsEye to control facial expressions, but the same technique can be used for other shape deformations (e.g., twisting, bending, etc.). An obvious alternative is to use free-form deformations to dynamically compute bending, twisting, and so forth [6]. Size (e.g., *large*, *small*) and aspect ratio (e.g., *flattened*, *squashed*) attributes are controlled by manipulating the 3D object's transform matrix.

4 The Depiction Process

All scenes are ultimately defined in terms of a set of low-level depictors (3D objects and their spatial and graphical properties). The job of WordsEye's depiction module is to translate the high-level semantic representation produced by the linguistic analysis into these low-level depictors. The process involved in the creation and application of depictors, leading to a final rendering, is outlined below:

1. Convert the semantic representation from the node structure produced by the linguistic analysis to a list of typed semantic elements with all references resolved.
2. Interpret the semantic representation. This means answering "who?" , "what?", "when?", "where?" "how?" when the actor, object, time, location, and method are unspecified.
3. Assign depictors to each semantic element.
4. Resolve implicit and conflicting constraints of depictors.
5. Read in referenced 3D models.
6. Apply each assigned depictor, while maintaining constraints, to incrementally build up the scene.
7. Add background environment, ground plane, lights.
8. Adjust the camera, either automatically (currently by framing the scene objects in a three quarters view) or by hand.
9. Render.

We now describe this process in more detail.

4.1 Depiction Rules

The output of the linguistic analysis is a semantic representation of the text. The semantic representation in turn consists of a list of semantic elements representing the various constituent meanings and relations inherent in the input text. The main semantic element types are ENTITY, ACTION, ATTRIBUTE and RELATION which roughly correspond to nouns, verbs, adjectives and prepositions. Some additional, more specialized types are PATHS, TIMESPEC, CONJUNCTION, POSSESSIVE, NEGATION, CARDINALITY. Each type of semantic element has various type-specific parameters. For example, ACTIONS have tense. We omit these incidental parameters in the examples below.

As an example, the sentence *The cowboy rode the red bicycle to the store* is represented by the following semantic elements:

1. Entity: *cowboy*
2. Entity: *bicycle*
3. Entity: *store*
4. Attribute:
   Subject: <*element 2*>
   Property: *red*
5. Action:
   Actor: <*element 1*>
   Action: *ride*
   Object: <*element 2*>
   Path: <*element 6*>
6. Path:
   Relation: *to*
   Figure: <*element 5*>
   Ground: <*element 3*>

In order to depict a sentence, the semantic elements must be made graphically realizable. This is done by applying a set of *depiction rules*. Depiction rules are tested for applicability and then applied to translate semantic elements into graphical depictors.

As an example, we examine the depiction rule for the action *kick*. We handle several cases. The first case is for large objects, where the kicked object should be depicted on the ground in front of the actor's foot. This happens when there is no specified path (e.g., we say *John kicked the car*, as opposed to something like *John kicked the car over the fence*) and when the size of the direct object is larger than some arbitrary size (for example, 3 feet).

The second case is for small objects where no path or recipient is specified. It uses the *kick object* pose and substitutes the kicked object for the stand-in object in the pose, placed above the foot. This would be used with sentences like *John kicked the football.* The football gets put in the air, just above the foot.

The third case is used when the actor kicks the object on a path (e.g., to a recipient). This might correspond to sentences like *John kicked the football to Mary.* The PATH-DEPICTOR used in this depiction rule specializes in depicting objects on paths.

Note that some depictors are marked as *tentative* because they are just defaults and are not inherent to the kicking action. Also note that the depiction rules described above are somewhat simplified; they can be made arbitrarily more complex to handle various special cases and subtleties.

DEFINE-DEPICTION-RULE ACTION *kick*

Case1: no PATH or RECIPIENT, DIRECT-OBJECT size greater than 3 feet

Pose: *kick*, ACTOR

Position: ACTOR directly behind DIRECT-OBJECT

Orientation: ACTOR facing DIRECT-OBJECT

Case2: no PATH or RECIPIENT, DIRECT-OBJECT size less than 3 feet

Pose: *kick object*, ACTOR, DIRECT-OBJECT

Case3: PATH and RECIPIENT

Pose: *kick*, ACTOR

Path: DIRECT-OBJECT between ACTOR's *foot* and RECIPIENT

Orientation: ACTOR facing RECIPIENT

Pose: *catch*, RECIPIENT [tentative]

Orientation: RECIPIENT facing ACTOR [tentative]

Position: ACTOR 10 feet from RECIPIENT in Z axis [tentative]

Position: ACTOR 0 feet from RECIPIENT in X axis [tentative]

In some cases, depending on the object in question, different poses may be applied to the same basic action. For example, different objects (e.g., baseball, soccer ball, frisbee, javelin) are thrown in different manners. The depiction rules are responsible for finding the most appropriate pose for the action, given the total context of the semantic elements.

For attributes, depiction rules can create depictors for size, color, transparency, aspect ratio, and other directly depictable properties of the objects. Sometimes an attribute is best depicted by attaching an iconic appendage to the object. This is illustrated in the following depiction rule, which is used to depict a *spinning* object. Since we cannot directly depict motion, we can suggest it iconically by putting a circular arrow above the given object.

DEFINE-DEPICTION-RULE ATTRIBUTE *spinning*
Spatial-Relation: *above*, SUBJECT, *circular arrow 3D model*

For entities themselves, depiction rules are normally responsible for selecting which 3D object to use. This is currently done by selecting randomly among those matching the given term. In some cases, however, an entity is depicted by an assembly of separate 3D objects. With environments (e.g., a living room), this will almost always be the case. And in another example, *cowboy* might be depicted as a human character wearing a cowboy hat. So we create a pose depictor that positions and attaches the cowboy hat to the actor's head.

DEFINE-DEPICTION-RULE ENTITY *cowboy*
Pose: *wear cowboy hat*, ACTOR

4.2 Implicit Constraints

In certain circumstances it is desirable to add depictors for constraints that are not explicitly stated, but rather are based on common sense knowledge or are in some way deducible from the semantic representation. A set of transduction rules is invoked to do this.

Consider: *The lamp is on the table. The glass is next to the lamp.* Although it is not stated that the glass is on the table, we probably want it there rather than floating in the air next to the lamp. To do this, we invoke a rule that says "If X is next to Y, X is not already on a surface, and X is not an airborne object (e.g., a helium balloon)" then "Put X on the same surface as Y".

Consider next the sentence *The cat and the dog are on the rug.* Since there is no specification of where on the rug the cat and dog are located, it would be logically consistent to put the dog and cat in exactly the same place on the rug. To avoid this unintuitive result, we invoke a rule that says "If X is on Y, and Y is on Z, and X and Y are not spatially constrained" then "Put X next to Y". Note that although the previous rule is specified only with respect to the *on* relation, a more general formulation is possible.

4.3 Conflicting Constraints

Depiction specifications sometimes conflict with one another. This typically occurs when the default depictors assigned by an action conflict with those explicitly specified elsewhere. For example, *John kicked the ball to Mary* will generate depictors to put John in a *kick* pose, put John behind and facing Mary, put the ball between John and Mary, etc. Some of those depictors, such as the exact positions of the two characters, are labeled *tentative* because they are just defaults and are not inherent to the kicking pose.

1. POSE: John in kick pose
2. PATH: Ball between John's foot and Mary
3. ORIENTATION: John facing Mary
4. POSE: Mary in catch pose [tentative]
5. ORIENTATION: Mary facing John [tentative]
6. POSITION: John 10 feet from Mary in Z axis [tentative]
7. POSITION: John 0 feet from Mary in X axis [tentative]

But assume we add the specifications that *Mary is 20 feet to the left of John* and *Mary is 30 feet behind John.* which generates these depictors:

8. POSITION: Mary 20 feet from John in X axis
9. POSITION: Mary 30 feet from John in Z axis

We now have a conflict between depictors 6,7 and 8,9. To resolve these, a transduction rule is invoked that when detecting a conflict between depictors X and Y, where X is tenative, will remove depictor X. So, in this example since depictors 6,7 are marked as tentative, they are removed. The following is the result:

1. POSE: John in kick pose
2. PATH: Ball between John's foot and Mary
3. ORIENTATION: John facing Mary
4. POSE: Mary in catch pose [tentative]
5. ORIENTATION: Mary facing John [tentative]
6. POSITION: John 20 feet from Mary in Z axis
7. POSITION: John 30 feet from Mary in X axis

4.4 Applying Depictors

In order to actually create a coherent scene, the various independent graphical depictors (poses, spatial relations, etc.) derived from the semantic representation need to be applied. This is done by applying constraints in a simple prioritized manner:

1. Objects are initialized to their default size and shape. Size and color changes to objects are made at this stage also.
2. Apply shape changes and attachments. Characters are put into poses, and instruments and other pose-related objects are attached. At the same time, shape changes (for facial expressions, etc.) are made. The integration of upper and lower body poses are also handled at this stage.
3. Once objects are in their correct shapes and poses, all objects are positioned, with the exception of objects placed on paths in step 5. Once objects are constrained together (independently in each axis), neither can be moved without the other (along that axis).
4. With objects in the correct poses/shapes and positions, orientations are applied. This handles cases where one object is specified to face another or in some particular direction.
5. Dynamic operations such as placing objects on paths and IK are now performed.

4.5 Interpretation, Activities, Environment

In order for text to be depicted, it must first be interpreted. A sentence like *John went to the store* is somewhat vague. We do not know if John drove, walked, ice skated, hitchhiked, etc. Furthermore, we do not know how old John is, how he is dressed, or whether he is doing anything else on the way. Nor do we know the type of store, its location, what type of building it is in, or the path taken to get there. To be depicted, these must be resolved. This will often involve adding details that were not explictly stated in the text.

We rely on the functional properties of objects to make some of these interpretations. Verbs typically range along a continuum from pure descriptions of state changes like *John went to the store* to more explicit specifications of manner like *John crawled to the store*. Sometimes an instrument (or vehicle) is mentioned as in *John rode a bicycle to the store* while in other cases, the type of instrument is only implied by the verb as in *John rode to the store*. To find implied instruments, we look for objects whose functional properties are compatible with the instrument type demanded by the verb. In this case we want a rideable vehicle and find (among others) a *bicycle*. We then apply the "usage" pose for that object (bicycle). In this way, the sentence *John rode to the store* gets depicted with John in a riding pose on a bicyle.

Very often the interpretation will depend on the setting of the scene, either an environment (e.g., a forest) or an activity (e.g., a football game). Sometimes there is no explicitly specified environment, in which case an environment compatible with the rest of the text could be supplied. Consider, for example, *The flower is blue*. Rather than just depicting a blue flower floating on the page, we have the option of supplying a background. The simplest case for this is a ground plane and/or supporting object. For more visually complex cases, we may want to put the flower in a vase on a fireplace mantle in the middle of a fully decorated living room. Even when the setting is specified as in *John walked through the forest*, it must be resolved into specific objects in specific places in order to be depicted.

It should be noted that the same type of semantic inferences made with instrumental objects can also be applied to settings. For the sentence *John filled his car with gas*, we know he is probably at a gas station and we might want to depict John holding the gas pump. WordsEye currently does not have enough real-world knowledge or the mechanisms in place to handle environments or activities but we recognize their importance both to interpreting semantic representations and adding background interest to otherwise more purely literal depictions.

4.6 Figurative and Metaphorical Depiction

Many sentences include abstractions or describe non-physical properties and relations, and consequently they cannot be directly depicted. We use the following techniques to transform them into something depictable:

Textualization: When we have no other way to depict an entity (for example, it may be abstract or maybe we do not have a matching 3D model in our database), we generate 3D extruded text of the word in question. This can sometimes generate amusing results: see Figure 14.

Emblematization: Sometimes an entity is not directly depictable, but some 3D object can be an emblem for it. In those cases, the emblem is used. A simple example of an emblem is a light bulb to represent the word *idea*, or a church to (somewhat ethnocentrically) represent *religion*. We also use emblems to represent fields of study. For example, *entomology* is depicted by a book with an insect as an emblem on its cover.

Characterization: This is a specialized type of emblematization related to human characters in their various roles. In order to depict these, we add an article of clothing or have the character hold an instrument that is associated with that role. So, for example, a *cowboy* will wear a cowboy hat, a *football player* will wear a football helmet, a *boxer* will wear boxing gloves, a *detective* might carry a magnifying glass, and so on.

Conventional icons: We use comic book conventions, like thought bubbles, to depict the verbs *think* or *believe*. The thought bubble contains the depiction of whatever is being thought of. Likewise, we use a red circle with a slash to depict *not*; see Figure 15. The interior of the circle contains the depiction of the subject matter being negated. This same sort of depiction process can be applied recursively. For example, for *John thinks the cat is not on the table*, the thought bubble contains a red-slashed circle which in turn contains the cat on the table. Alternatively, *John does not believe the radio is green* is depicted with the slashed circle encompassing the entire depiction of John and the thought bubble and its contents; see Figure 16. Similarly, comic book techniques like speed lines and impact marks could be used to depict motion and collisions.

Figure 14: *The cat is facing the wall.*

Figure 15: *The blue daisy is not in the army boot.*

Literalization: Sometimes figurative or metaphorical meanings can be depicted most effectively in a literal manner: see Figure 17. We note that this is a well established technique. For example, T. E. Breitenbach's poster "Proverbidioms"[3] contains depictions of hundreds of figures of speech. *Throwing the baby out with the bathwater* is depicted literally, as a baby being tossed out a window along with a tub of bathwater. This approach comes naturally to WordsEye.

Personification: When metaphorical statements are interpreted literally, an inanimate or abstract entity often needs to be depicted in a human role (e.g., *Time marches on*). Our current minimalist approach is to affix some representation (flattened object, text, or emblem) of that entity onto a generic human character's chest as a visual identifier, like Superman's "S". A more satisfactory solution would be, in cartoon style, to give the object a set of generic legs and arms and a superimposed face.

[3] `www.tebreitenbach.com/posters.htm`

Figure 16: *John does not believe the radio is green.*

Figure 17: *The devil is in the details.*

Degeneralization: General categorical terms like *furniture* cannot be depicted directly. We depict these by picking a specific object instance of the same class (in this case, perhaps *chair*). This works well enough in most cases, as in *John bought a piece of furniture*. But sometimes, the reference is to the general class itself and hence the class, not an instance of it, should be depicted as in *This table lamp is not furniture*. We currently do not handle this case. One depiction strategy might be to choose a representative, generic looking object within the class and affix a textual label consisting of the class name itself.

5 Discussion and Future Work

We believe WordsEye represents a new approach to creating 3D scenes and images. It is not intended to completely replace more traditional 3D software tools, but rather to augment them by, for example, allowing one to quickly set up a scene to be later refined by other methods. WordsEye focuses on translating the semantic intent of the user, as expressed in language, into a graphic representation. Since semantic intent is inherently ambiguous, the resulting 3D scene might only loosely match what the user expected. Such variability, however, will be an asset in many cases, providing interesting and surprising interpretations. And when users want to control a depiction more precisely, they can adjust their language to better specify the exact meaning and graphical constraints they envision. We believe that the low overhead of language-based scene generation systems will provide a natural and appealing way for everyday users to create imagery and express themselves.

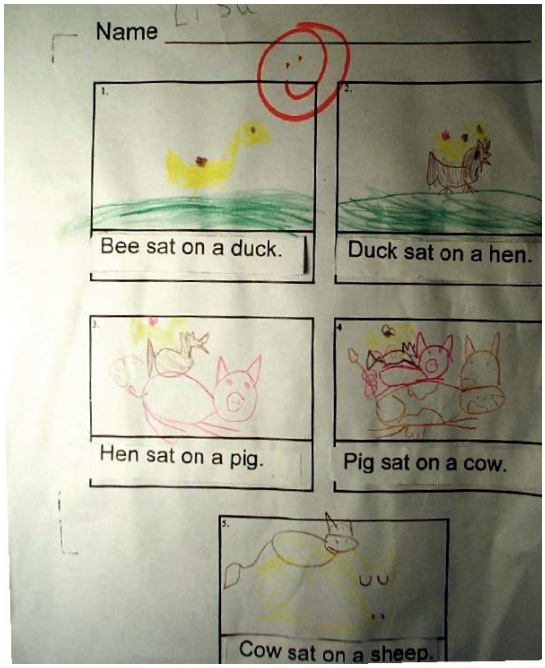

Figure 18: Some real 1st Grade homework, and a WordsEye "interpretation".

WordsEye is currently a research project and is under active development. We expect that eventually this technology will evolve into something that can be applied to a wide variety of applications, including: First and second language instruction (see Figure 18); e-postcards (cf. `www.bluemountain.com`); visual chat; story illustrations; game applications; specialized domains, such as cookbook instructions or product assembly.

In its current state, WordsEye is only a first step toward these goals. There are many areas where the capabilities of the system need to be improved, such as: Improvements in the coverage and robustness of the natural language processing, including investigating corpus-based techniques for deriving linguistic and real-world knowledge; language input via automatic speech recognition rather than text; a larger inventory of objects, poses, depiction rules, and states of objects; mechanisms for depicting materials and textures; mechanisms for modifying geometric and surface properties of object parts (e.g. *John has a long red nose*); environments, activities,

and common-sense knowledge about them; comic-style multiple frames for depicting sequences of activities in a text; methods for handling physical simulations of skeletal dynamics, shape deformation and natural phenomena. Work is ongoing to improve WordsEye along these various lines. However, we feel that even in its present state, WordsEye represents a significant advance in a new approach to the graphical expression of a user's ideas.

6 Acknowledgements

We thank Michael Collins, Larry Stead, Adam Buchsbaum, and the people at IZware for support of some of the software used in WordsEye. We also thank audiences at the University of Pennsylvania and Bell Labs for feedback on earlier presentations of this material. Finally we thank anonymous reviewers for SIGGRAPH 2001 for useful comments.

References

[1] *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, San Mateo, CA, 1995. Morgan Kaufmann.

[2] G. Adorni, M. Di Manzo, and F. Giunchiglia. Natural Language Driven Image Generation. In *COLING 84*, pages 495–500, 1984.

[3] N. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao, and M. Palmer. Parameterized Action Representation for Virtual Human Agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 256–284. MIT Press, Cambridge, MA, 2000.

[4] R. Bindiganavale, W. Schuler, J. Allbeck, N. Badler, A. Joshi, and M. Palmer. Dynamically Altering Agent Behaviors Using Natural Language Instructions. In *Autonomous Agents*, pages 293–300, 2000.

[5] C. Brugman. The Story of Over. Master's thesis, University of California, Berkeley, Berkeley, CA, 1980.

[6] Y. Chang and A. P. Rockwood. A Generalized de Casteljau Approach to 3D Free-Form Deformation. In *SIGGRAPH 94 Conference Proceedings*, pages 257–260. ACM SIGGRAPH, Addison Wesley, 1994.

[7] K. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Morristown, NJ, 1988. Association for Computational Linguistics.

[8] S. R. Clay and J. Wilhelms. Put: Language-Based Interactive Manipulation of Objects. *IEEE Computer Graphics and Applications*, pages 31–39, March 1996.

[9] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1999.

[10] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[11] C. Freksa, C. Habel, and K. F. Wender, editors. *Spatial Cognition*. Springer, Berlin, 1998.

[12] J. Funge, X. Tu, and D. Terzopoulos. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In *SIGGRAPH 99 Conference Proceedings*, pages 29–38. ACM SIGGRAPH, Addison Wesley, 1999.

[13] Sanda Harabagiu and Steven Maiorano. Knowledge-lean coreference resolution and its relation to textual cohesion and coherence. In *Proceedings of the ACL-99 Workshop on the Relation of Discourse/Dialogue Structure and Reference*, pages 29–38, College Park, MD, 1999. Association for Computational Linguistics.

[14] B. Hawkins. *The Semantics of English Spatial Prepositions*. PhD thesis, University of California, San Diego, San Diego, CA, 1984.

[15] A. Herskovitz. *Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English*. Cambridge University Press, Cambridge, 1986.

[16] R. Hudson. *Word Grammar*. Blackwell, Oxford, 1984.

[17] R. Langacker. *Foundations of Cognitive Grammar : Theoretical Prerequisites*. Stanford University Press, Stanford, CA, 1987.

[18] B. Levin. *English Verb Classes And Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL, 1993.

[19] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[20] P. Olivier and K.-P. Gapp, editors. *Representation and Processing of Spatial Prepositions*. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.

[21] C. W. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. In *SIGGRAPH 87 Conference Proceedings*, pages 25–34. ACM SIGGRAPH, Addison Wesley, 1987.

[22] G. Senft, editor. *Referring to Space: Studies in Austronesian and Papuan Languages*. Clarendon Press, Oxford, 1997.

[23] T. Winograd. *Understanding Natural Language*. PhD thesis, Massachusetts Institute of Technology, 1972.

[24] A. Yamada. *Studies on Spatial Description Understanding based on Geometric Constraints Satisfaction*. PhD thesis, Kyoto University, Kyoto, 1993.

[25] J. Zhao and N. Badler. Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures. *ACM Transactions on Graphics*, pages 313–336, October 1994.