July 2002

ACUMEN: Amplifying Control and Understanding of Multiple ENtities

Jan Allbeck
University of Pennsylvania

Karin Kipper
University of Pennsylvania

Charles Adams
University of Pennsylvania

William Schuler
University of Pennsylvania

Elena Zoubanova
University of Pennsylvania

See next page for additional authors

Recommended Citation

Allbeck, J., Kipper, K., Adams, C., Schuler, W., Zoubanova, E., Badler, N. I., Palmer, M., & Joshi, A. K. (2002). ACUMEN: Amplifying Control and Understanding of Multiple ENtities. Retrieved from http://repository.upenn.edu/hms/33

Postprint version. Copyright ACM, 2002. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Proceedings of the International Conference on Autonomous Agents*, pages 191-198.
Publisher URL: http://doi.acm.org/10.1145/544741.544788

ACUMEN: Amplifying Control and Understanding of Multiple ENtities

Abstract

In virtual environments, the control of numerous entities in multiple dimensions can be difficult and tedious. In this paper, we present a system for synthesizing and recognizing aggregate movements in a virtual environment with a high-level (natural language) interface. The principal com- ponents include: an interactive interface for aggregate con- trol based on a collection of parameters extending an exist- ing movement quality model, a feature analysis of aggregate motion verbs, recognizers to detect occurrences of features in a collection of simulated entities, and a clustering algorithm that determines subgroups. Results based on simulations and a sample instruction application are shown.

Comments

Postprint version. Copyright ACM, 2002. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Proceedings of the International Conference on Autonomous Agents*, pages 191-198.
Publisher URL: http://doi.acm.org/10.1145/544741.544788

Author(s)

Jan Allbeck, Karin Kipper, Charles Adams, William Schuler, Elena Zoubanova, Norman I. Badler, Martha Palmer, and Aravind K. Joshi

ACUMEN: Amplifying Control and Understanding of Multiple ENtities

Jan Allbeck, Karin Kipper, Charles Adams, William Schuler, Elena Zoubanova,
Norman Badler, Martha Palmer, and Aravind Joshi
Computer and Information Science
200 S. 33rd St.
Philadelphia, PA 19104-6389
allbeck@seas.upenn.edu

ABSTRACT

In virtual environments, the control of numerous entities in multiple dimensions can be difficult and tedious. In this paper, we present a system for synthesizing and recognizing aggregate movements in a virtual environment with a high-level (natural language) interface. The principal components include: an interactive interface for aggregate control based on a collection of parameters extending an existing movement quality model, a feature analysis of aggregate motion verbs, recognizers to detect occurrences of features in a collection of simulated entities, and a clustering algorithm that determines subgroups. Results based on simulations and a sample instruction application are shown.

Categories and Subject Descriptors

I.5 [**Computing Methodologies**]: Pattern Recognition; I.6 [**Computing Methodologies**]: Simulation and Modeling; I.2.7 [**Computing Methodologies**]: Natural Language Processing

1. INTRODUCTION

Human language developed partly from a need to compress and linearize human experience. The world we live in is multi-dimensional: three spatial dimensions plus time, as well as the activities of numerous natural, mechanical, physical, and sentient entities. Although we can capture experience through multiple sensors, we possess very few innate methods – mainly gestures or verbal expression – for conveying complete experiences to others. We thus augment the cognitive powers of others by compressing our experience into sketches and words. In a military environment, a commander relies on the communications of others to characterize and summarize complex, multi-entity movements and situations. A traffic reporter characterizes off-nominal vehicle flows purely through verbal summaries. During urban emergencies or civil unrest, the police attempt to characterize group movements to focus resources and control threats.

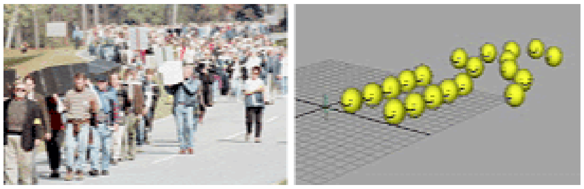

Figure 1: Aggregate Movement Examples

Our goal is to build efficient computational models for creating and recognizing aggregate activities within a collection of entities. We capture the cognition enhancing function of language to both recognize and describe actions of groups of entities. Such a capability allows scenario creators to effectively control visual simulations of aggregate entities such as in crowds or urban populations. It permits the compression and textual linearization of voluminous data of aggregate movements to establish process, context, and deviation points for decision-makers, and allow after action review of constructive, live, or virtual simulations.

Descriptions convey much information without requiring one to view particular imagery or animations; they can be conveyed by a straightforward text message and can be assimilated easily at the cognitive level: creating a mental picture of these dynamic situations is effortless for the hearer. For example, "The [funeral] procession walked in a slow but energetic line followed by a growing collection of people and was watched carefully by a milling crowd of onlookers." (See Figure 1).

Our approach is to decompose higher-level concepts, such as assembling and dispersing, into simpler features that may be quickly and robustly detected in large aggregate populations. The primary alternative would be to code specific high-level action recognizers, but this would be both cumbersome and difficult to scale. We specifically seek to take advantage of subject matter experts by using natural language to communicate aggregate activities, thus avoiding a direct translation of every interesting high-level concept into code. Work in defining such features is leading to useful computational models, grounded in cognitive movement understanding.

Creating movements of large numbers of independent entities such as crowds or urban populations is time-consuming

AAMAS'02, July 15-19, 2002, Bologna, Italy.
Copyright 2002 ACM 1-58113-480-0/02/0007 ...$5.00.

and costly to modify. A natural language based constructive interface enhances the generation and modification of aggregate behaviors. A natural language based recognition and description module permits compressing and linearizing huge amounts of movement information, thus amplifying cognitive understanding of context and streamlining its presentation to decision-makers. We are not claiming that visual presentation of such data is not useful or valid; rather, our claim is that the linearization provided by language can be a useful adjunct in delivering compressed information to a decision-maker. Since language may be verbalized as well as read, and since auditory delivery of such information can be used as an alert, these automated descriptions would not require the constant vigilance of direct visual monitoring. Auditory alerts for specific events are a staple of, for example, aircraft control systems.

This paper introduces *ACUMEN*, which builds on our work on connecting natural language and animation. We begin by describing our *Parameterized Action Representation* and the system that processes it. We then discuss aggregate entities, their movements, and related research. Next, we present a detailed description of the *ACUMEN* system and a demonstration of the system. Finally, we present our conclusions and discuss future extensions.

2. PREVIOUS WORK

Any dynamic situation must be created by some (possibly unknown) processes, so any description or recognition of that situation must utilize a process representation to account for and interpret dynamic data. We have developed a process representation called the *Parameterized Action Representation* (PAR) [6]. A primary component of the system that processes PARs for animation is the *Actionary*$^{\mathrm{TM}}$. It contains persistent, hierarchical databases of agents, objects, and actions. The agents are treated as special objects and stored within the same hierarchical structure as the objects. Parameterized actions are represented in a frame-like structure with fields that may be used to generate the actions on or with a given agent or set of agents. Because animation and natural language have different constraints we have designed independent hierarchies for the representation of actions. On the one hand, we have natural language *PAR schemas* that are derived from the idea that verbs share common semantics and can be described by a set of semantic predicates associated with arguments (which are participants in the action) and selectional restrictions on these arguments. This conjunction of semantic predicates captures the semantics of a verb or a class of related verbs. On the other hand, the hierarchy for animation is derived from motion semantics based on the requirements of the animation. An uninstantiated PAR (UPAR) is essentially a definition for an action, containing only default properties for the action. When an action becomes associated with an agent, it is called an instantiated PAR (IPAR) and contains specific information about the agent, objects, and other properties. All the UPARs are stored hierarchically within the *Actionary.*

PARs can be stored, modified, interpreted, and transferred like data packets, and are generally used as dynamic information objects. We have extended our representation and our software to handle both individual and aggregate actions so that it can be used in applications for decision-making. More details on PAR and the processing of actions can be found in [6, 4, 3].

3. AGGREGATE ENTITIES

Things that move do so over time. The activities that they engage in are fluid and changing. Often we perceive an action only as it crystallizes out of other random or chaotic activities. *Dispersing*, for instance, only becomes apparent after the process has progressed for some time. Often it is easier to detect that something has happened by seeing an end or termination, such as an *escape* from a place. Human language has developed lexical items with detailed semantics to characterize such processes. Effective and efficient characterizations of such spatio-temporal, ongoing processes present an exciting opportunity for augmenting the cognitive space available to decision-makers [20].

Much work has been done on the generation and recognition of individual actions [7, 17, 5, 22]. Bindiganavale's work [5], in particular, recognized human actions and described them using our Parameterized Action Representation.

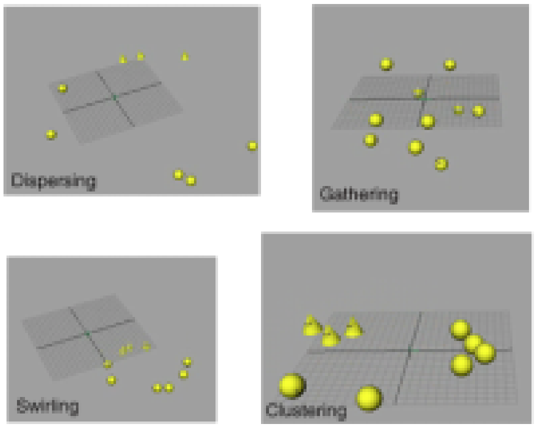

Figure 2: Aggregate Movement Synthesis Examples.

Aggregate movement generation has also been a classic computer graphics research topic. Reynolds [19] created a particle system based simulator for flocking, herding, and schooling. The gross movement of the aggregate is controlled by a *migratory* attractor. The overall motion of the simulated flock is the result of simple behaviors of the individual simulated birds. More recently, Musse and Thalmann [18] have reported a method of crowd simulation with various levels of autonomy. Their work incorporated rule-based behaviors with programmed and user controlled agents in the same scenario. The movement of these crowds or groups is based on *interest points*, which act as attractors for the crowds. Aggregate simulation systems such as these have limited control over the movement of the aggregates. Dispersion of a group from a point in this way would be quite difficult as an attractor would have to be created for each individual entity. Our aggregate simulation system uses attractors in combination with repulsors, randomization, and higher level parameterization to achieve varied, realistic aggregate movement behaviors.

Biographic Technologies has created a Maya$^{\mathrm{TM}}$ plug-in for autonomous character control for crowds. However, an animator still has to write the rules to make each character behave as desired [14]. Our system, on the other hand, is based on lexical semantics and has a natural language interface which allows non-animators to control the characters. Additionally, our system synthesizes a greater number of aggregate behaviors and includes recognition of aggregate movements.

Many military simulations, such as [10], use computer generated forces but treat aggregates as a single entity, which only become disaggregated when they enter a conflict zone.

A *Capture the Flag* simulator was developed by Paul Cohen and his group in the Experimental Knowledge Systems Laboratory at the University of Massachusetts at Amherst [9]. In this war-gaming environment, aggregate entities are represented as *blobs*. Units can change shape and adopt columnar and frontal formations, as well as wedge and "V" formations or can change shape in response to terrain features. Marsella and Johnson [15] created a system to aid instructors in the evaluation of military team training through high-level assessment of simulations based on situation spaces. This work includes the evaluation of aggregate formations, but the evaluation is done on a small number of formations with few entities. We have found no research in computer generated forces and military simulations that recognizes aggregate movements with the robustness and scale of our system.

Recognition of aggregate movements has been done for commentaries in RoboCup soccer simulations [2]. These commentator systems take position and orientation information along with game score and play modes and create higher level conceptual units that can be used to produce natural language commentaries. The scene interpretation or recognition is, unlike ours, highly domain specific.

4. SYSTEM SOLUTIONS

Cognitive understanding of a dynamic situation depends on understanding the processes, flows, and changing spatio-temporal processes at work. Representing and identifying dynamic processes in 3D space is key to presenting cognitively manageable chunks of information. PARs form an abstraction of the observed events and processes; they can then be monitored and reported or queried as the application requires. This representation to be extended to support descriptions of aggregates as a situation evolves. Key transitions in descriptions signal context changes. Natural language can be used to describe transitions, terminating points, processes, etc. Moreover, natural language includes qualitative terms (adjectives and adverbs) which focus on degree, rate, focus, or manner. These terms summarize highly complex aggregate activities by filtering out individual details and exposing emergent behaviors.

PARs were designed to represent actions, so its syntax takes into account the relevant information for such a task. It is well-suited to describe any sort of process, whether human, machine, or aggregate. Both military and civil environments may require that a decision-maker understand dynamic states and infer intention, control and threat: all of which have a strong spatial component. PAR is the basis for building and describing situation states with parameterized models.

We have refined our PAR system to make it more robust and able to represent aggregates and their movements. The new system incorporates an aggregate movement synthesizer and an aggregate movement recognizer.

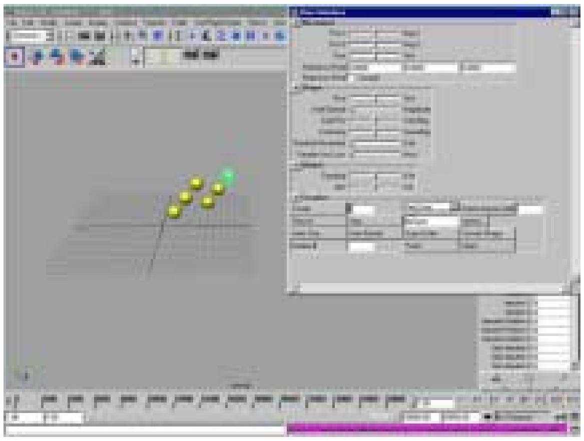

Figure 3: Generation Interface: This is an interface to the aggregate motion synthesizer.

4.1 Aggregate Movement Synthesis

Creating movements of large numbers of independent entities such as crowds or urban populations is time-consuming and costly to modify. A natural language-based interface enhances the generation and modification of aggregate behaviors. The crowd behaviors in the *Institute for Creative Technologies'* virtual trainer demonstration [21] cannot be changed without extensive recoding. Our system permits rapid development of alternative, spatial, context-sensitive crowd behaviors. Though our system can support a natural language interface, we have created a GUI (Graphical User Interface) to the aggregate action synthesis portion of our *ACUMEN* system (Figure 3). The controls in this GUI are based on the semantics of terms used to describe movements of oriented entities.

Our aggregate motion simulator has been created within the scripting environment of Alias—Wavefront's graphics program, Maya^{TM}. Based largely on a particle system-like model of group simulation, this script uses dynamic forces acting on rigid bodies to produce the desired movement. Unlike particles, rigid bodies allow for greater control over individual orientation and accurate collision detection. The script maps lexical terms to lower level features, which are in turn mapped to dynamic primitives, such as attractors, repulsors, velocity, inertia, and randomization. The aggregate movement features are also used for aggregate motion recognition and are presented in Section 4.2.

Our synthesis module is able to demonstrate different types of aggregate movements, such as dispersing and gathering as exemplified in Figure 2. The synthesis module drives not only the visual display shown, but also stores a frame by frame accounting of the simulation as it is produced. This accounting contains the voluminous geometric information that is condensed by the recognition module of *ACUMEN*.

4.2 Aggregate Movement Recognition

Though the synthesis and recognition systems share a common data representation, PAR, the *ACUMEN* aggregate movement synthesis module and the movement recognition module are separate. The recognition system is able to be driven by any simulator that can convey geometric or spatial information. Geometric data from actual sensors can also be used. It recognizes instances of aggregate movements such as group dispersing and generates compact lexical descriptions of them, which are passed to other decision support or display systems.

We base the recognition module on computational definitions of features of terms, words, and concepts that describe movements of aggregate entities. By starting with low level features we are able to create a system that can be extended to new terms and refined for better recognition. The recognition of new terms only requires determining which semantic features are present for the new term. If we were to base the recognition of aggregate movement terms solely on individual computational models of each lexical item, the recognition of new terms could require extensive recoding. Our examination of features is based on both lexical semantics and movement observation science.

4.2.1 Lexical Semantics and Feature Selection

VerbNet [11] is a verb lexicon being built at the University of Pennsylvania exploiting the idea that verbs can be grouped into classes according to syntactic commonalities and shared semantic components. It uses Levin verb classes [13] to systematically construct lexical entries. These classes are based on the ability or inability of a verb to occur in pairs of syntactic frames called diathesis alternations. The sets of syntactic frames associated with a particular Levin class are supposed to reflect underlying semantic components that constrain allowable arguments and adjuncts.

Verb classes allow us to capture generalizations about verb behavior. This reduces not only the effort needed to construct the lexicon, but also the likelihood that errors are introduced when adding a new verb entry. Each verb class in VerbNet lists the thematic roles that the predicate-argument structure of its members allows, and provides descriptions of the syntactic frames corresponding to licensed constructions, with selectional restrictions defined for each argument in each frame. Each frame also includes semantic predicates describing the participants at various stages of the event described by the frame. Verb classes are hierarchically organized, ensuring that each class is coherent – that is, all its members have common semantic elements and share a common set of thematic roles and basic syntactic frames.

PAR schemas, a component of the Actionary^{TM}which specifies natural language semantics for actions, derive from VerbNet the idea that verbs can be represented in a lattice that allows semantically similar verbs, such as motion verbs or verbs of contact, to be closely associated with each other under a common parent that captures the properties these verbs all share. The highest nodes in the hierarchy are occupied by generalized PAR schemas which represent the basic predicate-argument structures, the lower nodes are occupied by progressively more specific schemas that inherit information from the generalized schemas.

These action schemas are based on VerbNet entries and are described by a set of semantic predicates associated with arguments (participants in the action) and selectional restrictions on these arguments. The semantic predicates used to describe an action can be viewed as basic features (e.g. motion, contact), and their conjunction captures the semantics of a verb or a class of related verbs. Figure 4 shows an example representation for the verb 'to shove'. First, there is the syntactic frame (ARG0 verb ARG1), corresponding to the transitive use of the verb, this also serves to establish the minimal set of participants in the action. The selectional restrictions for the arguments are captured by the predicates *is_animate(ARG0)* and *is_concrete(ARG1)*. The other predicates in the conjunction describe the semantics of the event *Agent shoves Patient*.

```
shove / ARG0-v-ARG1
      / is_animate(ARG0)
        is_concrete(ARG1)
        contact(during(e),ARG0,ARG1)
        exert_force(during(e),ARG0,ARG1)
        motion(during(e),ARG1)
        cause(ARG0,e)
```

Figure 4: PAR **schema for 'shove'**

Additionally, our action descriptions are based on features of human movement observation science. Originated by Rudolf Laban [12], Laban Movement Analysis (LMA) today is a creative method of movement study for observing, describing, notating, and interpreting human movement. LMA provides insights into one's personal movement style and increases awareness of what movement communicates and expresses. Chi et al. [8, 23] built a system called EMOTE based on LMA to parameterize and modulate action performance. EMOTE is not an action selector per se; it is used to modify the execution of a given behavior and thus change its movement qualities or character. The power of EMOTE arises from the relatively small number of parameters that control or affect a much larger set. The EMOTE work focuses on the Effort and Shape components of LMA, because these two are the major direct specifications or indications of expressive human movements. Effort comprises four motion factors: Space, Weight, Time, and Flow. Each motion factor is a continuum between two extremes: (1) *indulging* in the quality and (2) *fighting* against the quality. Shape changes in movement can be described in terms of three dimensions: Horizontal, Vertical, and Sagittal.

After extensively analyzing the individual behavior of verbs that describe activities of aggregates to determine features that span a broad and expressive action space, and deciding to adopt the idea of verb classes used in VerbNet, we proceeded by grouping sets of these aggregate verbs into classes, extending the EMOTE features to group movement. Since EMOTE was designed for human arm gestures, the features had to be revised for aggregate entity movements. By using the EMOTE features as primitives we were able to capture both generalizations and distinctions among sets of verbs as shown in Table 1. Examples of verb classes created include verbs of gathering, dispersing, and milling.

We have found that the Effort dimensions (slow-fast; sudden-sustained; direct-indirect; free-bound) have meaningful correlation to aggregate behavior. We have also found that two of the Shape dimensions (advancing-retreating; spreading-enclosing;) correlate to aggregate shapes. Other dimensions of the study seem less appropriate (rising-sinking; left-right). Most aggregates entities, unlike individual humans, do not have an inherent top and bottom or left and right. Other factors, such as the focus of attraction are geometric features that have been found crucial to the proper characterization of group actions.

Figure 5 shows a PAR schema for 'to assemble' as in *The kids assemble (in a location)*, which has only one participant

	Gathering	Dispersing	Obj. Referential	Formation	Milling
Shape Advancing					
Retreating					
Spreading		x			
Enclosing	x				
Effort Slow					
Fast					
Sudden					
Sustained					x
Direct	x	x	x	x	
Indirect					x
Free					
Bound					
Other Obj Referent			x		
Structured				x	

Table 1: Feature Table

```
assemble    / ARG0-v
            / is_concrete(ARG0)
              is_plural(ARG0)
             !together_group(start(e),ARG0)
              transl_motion(during(e),ARG0)
              shape_enclosing(during(e),ARG0)
              effort_direct(during(e),ARG0)
              together_group(end(e),ARG0)
```

Figure 5: PAR **schema for 'assemble'**

(a plural concrete entity). As can be seen from the example entry, many of semantic predicates are taken directly from the EMOTE features. Our verb semantics allows for the decomposition of the action into stages similar to that of [16]. Semantic predicates hold true at different stages *(start, during, end)* of the event. The semantics of 'assemble' establish that at the start of the event the participants are not together as a group, during the event the participants move, the shape of the group is enclosing and the movement is direct, at the end of the event the participants should be together.

Although several verbs within a class are quasi-synonyms and can be described by the same set of features, it is also the case that we need more specific features to distinguish between some members of the same class. One the features have been computed, classes are determined and lexical descriptions generated. These compact descriptions can be passed to decision-making aids, used for operation summaries, or stored as essential context information.

4.2.2 Feature Calculation

Each of the features is mathematically computable based on the geometric and spatial data of the individuals and environment over time. Likelihoods or strengths of each of the features are computed. Many of the features are not boolean, such as nearness, which is a fuzzy value. The features apply to sets of individuals, so at any time there may be a number of individuals displaying a feature and a number of them not. A description of these features is given below:

- Advancing: constant average velocity vector
- Change of Direction: 10 degree or greater change in average velocity vector
- Spreading: increasing average distance from center of mass
- Enclosing: decreasing average distance from center of mass
- Slow: average velocity below user-specified threshold
- Fast: average velocity above user-specified threshold
- Sudden: orientation: rate of change of orientation above threshold and velocity: rate of change of velocity above threshold
- Random: high variance in individual orientation (corresponds to directness).
- Orientation Bound: uniform individual orientation and all individual positions aligned along orientation vector
- Position Bound: zero net change in position relative to world

4.2.3 Recognition Implementation

Figure 6 shows a diagram of the aggregate movement recognition system. Geometric data enters the recognition system from a simulator or sensor processing system. The data is stored so that time-segmented chunks of data can be processed. These data chunks are sent to the *Attribute Recognizer* and *Group Analyzer*. The *Attribute Recognizer* calculates the features or attributes and creates a histogram for testing and display. It also sends the attribute probabilities to the *Verb Recognition* component where they are used to chose terms to describe the aggregate movement.

Currently, the *Group Analyzer* uses K-means clustering [1] to determine subgroups of aggregates based on their movements. The clustering components include position, orientation, and velocity. When the algorithm begins two subgroups are assumed to exist in the population; each entity is randomly assigned to one of the two groups. The centroid of each group is calculated from the normalized components of each entity in the groups. Next, each entity is reassigned to a group based on the minimum distance between it and

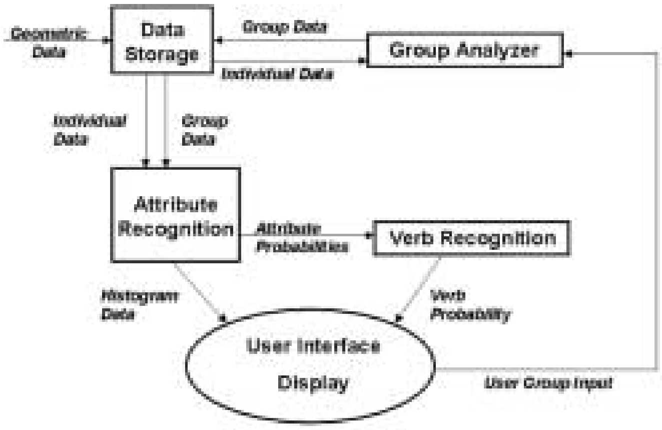

Figure 6: Diagram of the Aggregate Movement Recognition System

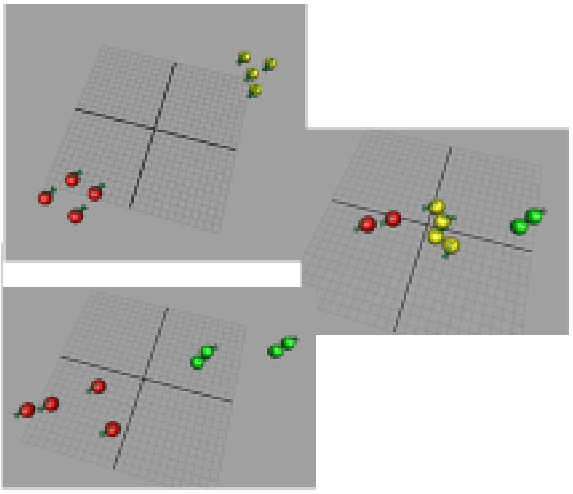

Figure 7: Example of K-means classification into subgroups.

the centroids of the groups. The centroids are recalculated and the process is iterated. The algorithm terminates when during an iteration no entities changed groups. In order to determine if more than two groups exist, the greatest distance between an entity and its group centroid is compared to a threshold value. If the distance is greater than the threshold, the same algorithm is run based on three groups (centroids). Preliminary results find this method effective (See Figure 7).

5. DEMONSTRATION

The computational definitions of verbs that describe movements of oriented entities are used in a scenario in to *guide* (program) the activities of aggregate entities in a simulation. Our demonstration also shows how these definitions may be used to *recognize* significant activities of a set of entities. The scenario takes place in a schoolyard with eight children and a supervising teacher (see Figure 8). Video clips of the demonstration can be found at
http://hms.upenn.edu/software/ACUMEN/demo.html

The scenario begins with the children milling around the yard. Additional instructions can be given to either the children or the teacher. Our video sample demonstrates the following natural language instructions:

Instructions given to the children:

1. When the teacher blows the whistle, disperse.
2. When the teacher opens the door, assemble.
3. When the teacher waves, gather in front of her.
4. If a skunk enters the playground, panic.

Instructions given to the teacher:

1. If the children are surrounding Ralph, blow a whistle.
2. If the children gather around the door, open it.
3. If the bell rings, assemble the children.

The actions in these instructions fall into three different categories: actions of an individual, individual actions given to an aggregate, and actions of an aggregate. Actions such as *blowing a whistle* or *opening a door* are included in instructions for an individual and performed only by that individual. Actions such as *panicking* are included in instructions for an entire group, but are performed separately by each member of the group. Finally, actions such as *gathering*, *assembling*, *dispersing*, *surrounding*, and *milling* can only be performed by aggregate entities and are the focus of this research.

The aggregate actions are divided into classes based on their features according to Table 1. Gathering movements have an enclosing shape and direct effort, which means that the density of the aggregate is increasing and the movement has a focus[1]. *Assembling*, *congregating*, and *getting together* are quasi-synonyms of *gathering*. Our classification scheme allows us to treat these terms as synonyms but, if required, do a more detailed classification. Dispersing movements are similar to gathering movements except that the shape is spreading instead of enclosing. *Dissipate*, *scatter*, and *spread out* have meanings similar to *dispersing*. Object Referential actions such as *surrounding* and *encircling* syntactically and semantically require an object which is the focus of the action. Unlike *dispersing* and *gathering*, which can have an implicit focus, object referential actions require an explicit focus. Milling actions are sustained actions lacking focus. These actions progress over a period of time in a wandering or meandering fashion. Some similar concepts in the milling class can be further distinguished by other effort parameters. *Bustling*, for example, implies a faster movement than *milling*. Formations are aggregate actions with structure, such as lines or columns. We have determined ways to recognize this structure in aggregate movements based on the position and orientation of the individuals. It is, however, difficult to generate these formations in a general way. As the generation and recognition of formations is very applicable to military domains, this remains a focus of our research.

[1] this focus can be explicit, as in *gather around the door* or implicit, as in *gather together*

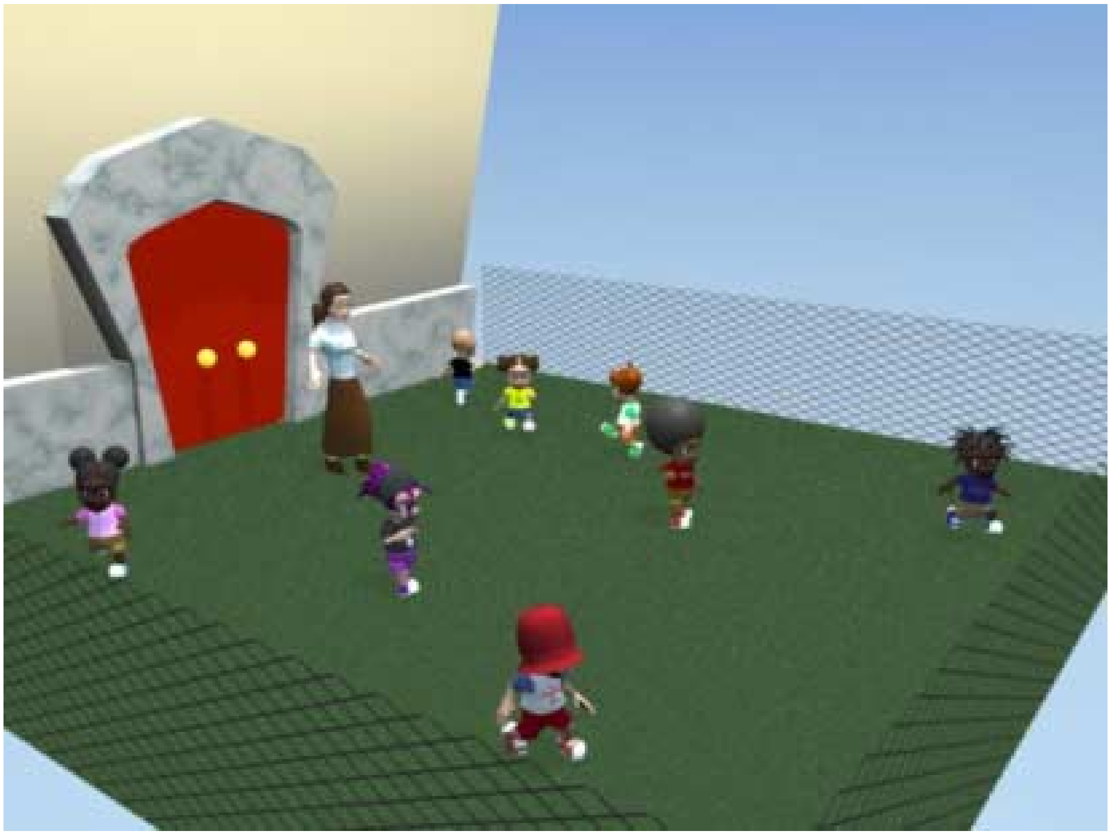

Figure 8: The Schoolyard Environment.

This demonstration also illustrates our system's ability to both synthesize and recognize aggregate movements. Instructions given to the children synthesis aggregate movements. The first two instructions given to the teacher are conditioned by the movements of the children and therefore only apply if a certain aggregate movement is recognized. The last instruction to the teacher illustrates our system's planning capabilities. When the bell rings, the teacher must assemble the children, which means that she must realize that a *waving* action will gather the children in front of her. This connection is made through the semantic information stored as parameters of the actions.

6. CONCLUSIONS

Amplifying a person's understanding of a complex multi-entity spatio-temporal situation is most properly served by a representation that understands spatial and directional terms and parametric modifiers. Process representations in PAR address recognition of individual or aggregate movements, manner or style of movement, and, will, in general, fundamentally support both retrospective (after action review) and prospective (predictive) analyses and responses. Our *ACUMEN* system uses the PAR representation to capture the semantics of aggregate movement for generation and recognition. The *ACUMEN* system synthesizes and recognizes movements of multiple entities in digital simulations and describes their movements in language terms. Unlike other research, the recognition of group actions is through feature-based action semantics based on verb classification and human movement observation science. We have found no research in computer generated forces and military simulations that recognizes aggregate movements with the robustness and scale of our system.

The more general goal of this research is to augment cognition by creating a compact description of aggregate entity movements which can then be used for human or automatic decision-making. Our scenario demonstrates these capabilities, as an illustration of a possible application. We are currently performing studies to determine the level of compression a natural language based representation provides and how this information may be best presented to decision-makers for both rapid absorption and long-term recall.

Our extensions include research in synthesis and recognition of simulated crowds and riots, as well as, military formations; and on the scalability of our design to large groups. In order to obtain real time recognition of large entity populations, we need to both sample the entities and to select those features most predictive of the activities of interest. For large crowds, it is neither feasible nor necessary to model all of the individuals in the crowd in order to analyze its behavior. More efficient strategies use sampling of sets of individuals, forming an approximate analysis of the different activities occuring, and then analyzing more entities to determine more precisely the boundaries between the different groups. Spatial zones may be used to sample geographically large or population dense areas; a subset of individuals from each spatial zone can be profiled. For very large populations this will tend to statistically select aggregate actions.

Although we have constructed feature matrices, it is hard to know *a priori* exactly which features of entities and relationships between entities will be most predictive of different activities. We need to incorporate the use modern feature selection techniques to make this determination, providing an ordering for the computation and analysis of the features. Finally, we are working toward a more fine-grained distinction for semantically related verbs within the classes, in order to further distinguish between them.

The *ACUMEN* system provides easy control of aggregate entities for simulations of military exercises, crowds, and urban environments. Through the use of an aggregate movement recognizer, a feature-based recognition procedure, and verb classification scheme, we have constructed a system for characterizing and summarizing complex, multi-entity movements in natural language terms. The resulting compressed information may be used for amplifying understanding of a situation and thereby aiding decision-making, in facilitating descriptions of multiple entity actions for after action reviews of real-time interactive simulations, or for capturing attention to significant group events with auditory alerts.

7. ACKNOWLEDGMENTS

We would like to acknowledge Salim Zayat (http://www.salimzayat.com/), who created the models and some of the animations for our video. We thank Koji Ashida and Michael Moore for their participation. We would also like to thank Alias—Wavefront for generously providing us with Maya^{TM}. This research is partially supported by NASA NAG9-1279, NSF CISE IIS99-00297 Actionary, and NSF IIS-9800658 VerbNet.

8. REFERENCES

[1] K. Alsabti, S. Ranka, and V. Singh. An efficient K-means clustering algorithm. In *First Workshop on High-Performance Data Mining*, 1998.

[2] E. Andre, K. Binsted, K. Tanaka-Ishii, S. Luke, G. Herzog, and T. Rist. Three robocup simulation league commentator systems. *AI Magazine*, 21(1):57–66, 2000.

[3] N. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao, and M. Palmer. A parameterized action representation for virtual human agents. In *Embodied Conversational Agents*, pages 256–284. MIT Press, 2000.

[4] N. Badler, M. Palmer, and R. Bindiganavale. Animation control for real-time virtual humans. *Comm. of the ACM*, 42(8):65–73, Aug. 1999.

[5] R. Bindiganavale. *Building Parameterized Action Representations for Observation.* PhD thesis, University of Pennsylvania, 2000.

[6] R. Bindiganavale, W. Schuler, J. Allbeck, N. Badler, A. Joshi, and M. Palmer. Dynamically altering agent behaviors using natural language instructions. In *Autonomous Agents 2000*, 2000.

[7] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, W. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *SIGGRAPH '94*, pages 413–420. ACM, 1994.

[8] D. Chi, M. Costa, L. Zhao, and N. Badler. The emote model for effort and shape. In *Proc. ACM SIGGRAPH*, pages 173–182, New Orleans, LA, 2000.

[9] B. Heeringa and P. Cohen. An underlying model for defeat mechanisms. In *Proceedings of the 2000 Winter Simulation Conference*, pages 933–939, 2000.

[10] C. Karr, R. Franceschini, K. Perumalla, and M. Petty. Integrating aggregate and vehicle level simulations. In *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*, pages 231–239, 1993.

[11] K. Kipper, H. T. Dang, and M. Palmer. Class-based construction of a verb lexicon. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, July-August 2000.

[12] R. Laban. *The Mastery of Movement.* Plays, Inc., Boston, 1971.

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

[14] G. Maestri. Autonomous character plug-in: Biographic technologies brings crowd control to maya. *Computer Graphics World*, 24(10):57, 2001. www.biographictech.com.

[15] S. Marsella and W. Johnson. An instructor's assistant for team-training in dynamic multi-agent virtual worlds. In *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems*, number 1452 in Lecture Notes in Computer Science, pages 464–473, 1998.

[16] M. Moens and M. Steedman. Temporal Ontology and Temporal Reference. *Computational Linguistics*, 14:15–38, 1988.

[17] F. Multon, L. France, M. Cani-Gascuel, and G. Debunne. Computer animation of human walking: a survey. *Journal of Visualization and Computer Animation*, 10:39–54, 1999.

[18] S. Musse and D. Thalmann. Hierarchical model for real-tim simulation of virtual human crowds. *IEEE Trans. on Visualization and Computer Graphics*, 7(2):152–164, 2001.

[19] C. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proc. SIGGRAPH '87*, volume 21, July 1987.

[20] J. Schraagen, S. Chipman, and V. Shalin, editors. *Cognitive Task Analysis*, page 18. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

[21] W. Swartout, R. Hill, J. Gratch, W. L. Johnson, C. Kyriakakis, C. LaBore, R. Lindheim, S. Marsella, D. Miraglia, B. Moore, J. Morie, J. Rickel, M. Thiebaux, L. Tuch, R. Whitney, and J. Douglas. Toward the holodeck: Integrating graphics, sound, character and story. In *Proc. Autonomous Agents '01*, pages 409–416, Montreal, 2001.

[22] M.-H. Yang and N. Ahuja. *Face Detection and Gesture Recognition for Human-Computer Interaction.* Kluwer Academic Publishers, 2001.

[23] L. Zhao. *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures.* PhD thesis, CIS, University of Pennsylvania, 2001.