SIMULATION AS ANIMATION

Craig Caldwell
School of Art and Design
Northern Arizona University

Thomas E. Linehan
Computer Visualization Department
Texas A&M University

Richard E. Parent
Computer Science Department
Ohio State University

ABSTRACT

Throughout the history of art, creative advances have paralleled technological advances in media, permitting the artist to illuminate new visual concepts. Such a situation exists today with the computer's assistance in synthesizing knowledge from different disciplines. Physically-based simulation permits such an integration and provides a new animation approach for the artist. This paper outlines a functional foundation for artists to use different simulation systems in the design of computer animation. The functional model seeks to bridge the qualitative - often idiosyncratic - conceptual orientation of the artist with the quantitative orientation of computer simulation.

INTRODUCTION

Computer animation today is still primarily mainly based on the traditional animation approach of specifying each key-frame manually. The computer assists in generating the needed frames "in between" these key-frames. This is not only tedious and complicated, but it is also limiting. It *conditions* the animator to think in terms of creating what a key-frame system can reasonably handle. This mind-set results in a tendency to dwell on 2D visual possibilities and character qualities that mimic traditional methods of generating animation.

As a result computer animation suffers from an absence of intriguing and expressive motion. To a certain extent this stems from the fact that certain phenomena are too visually or temporally complex to be adequately reproduced by the artist's visual skills alone. The temporal problems of creating animation - even within the computer environment - continue to exist: (a) generation of realistic motion is difficult, and (b) animating large collections of objects or figures that appear to interact is very complex and generally avoided. What is needed is a method for generating motion which is analogous to how motion is generated in the real world.

The ***most viable alternative*** is the adaptation of computer "simulation" techniques to an animation system. Computer simulations offer the hope of creating significant complex motion through the incorporation of behaviors of an object as it responds to its environment. Simulation techniques expand the boundaries of the visual process so that the physical limitations of an animator's time or the complexity of the idea does not have an overbearing influence on the creative outcome.

BACKGROUND

Using physical laws to generate animated motion is not a new idea. Traditional animators have long observed mechanical systems to obtain more life-like realism. Today, with the computer, we can generate realistic motion through the use of dynamic laws (Figure 1) or kinematic descriptions. These areas have been

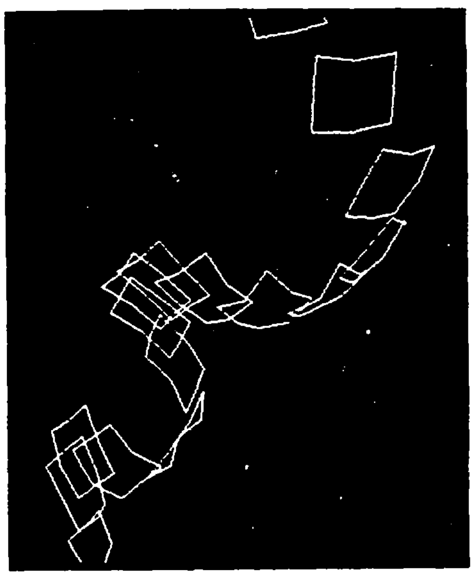

Figure 1, Dynamic simulation of a flexible object by Dave Haumann.

pioneered by scientists such as Haumann[1], Barr[2], Wilhelms[3], Zeltzer[4], Thalmann[5], and Girard[6]. The resultant systems have, in general, been research oriented and have focused on a specific method (i.e. dynamics versus kinematics). The individual limitations of each system, however, indicates a need for a broader outlook, one which can accommodate the changing needs of artists. This paper seeks to propose a functional model that encompasses the implementation of physically-based simulation as an animation option for the artist. The functional model seeks to bridge the qualitative - often idiosyncratic - conceptual orientation of the artist with the quantitative orientation of computer simulation. This in turn will influence the creation of more subtle and realistic animation for sophisticated viewers.

From these new simulation/animation hybrid systems new operands will evolve that tap the unexplored realms of the computer medium.

> The term "simulation", itself, rather than "animation", denotes a shift in control from the animator to the underlying physics of the environment. One would like a system for specifying motion which combines the realism of dynamic simulation without removing control from the animator.[7]

It is hypothesized that the re-creation of motion in the computer has far-reaching ramifications for the animator. In an effort to arrive at a functional mode this paper addresses "How should such a physically-based simulation system be structured for its use in animation?" and "How does the proposed system extend the existing means or create tools for the animator?"

It was determined that to create this functional model from the strictly subjective perspective of artists was not feasible. This would have dictated individual programs every time a new variation needed to be played out. It became obvious that the only way to structure the model was on a systems-oriented approach.[8] The applicability of artistic initiatives differentiates this model from the current computer simulations and their applications.

This functional model is successively organized in concentric layers. Each layer contains abstractions joined through a logical interaction or interdependence. Layering provides a powerful structure from which to create animations governed by physical laws and driven by our imagination. Such a structure permits the artist to transcend various levels of detail.

From this functional model the interrelationships of particular simulation components and the causes of changes in the simulation can be understood. To reproduce the physical interactions of our world it is necessary to view the salient features of this model as a set of ***geometric primitives*** (how structure is represented in the computer), ***mechanical attributes*** (the characteristic qualities of an object's movement), ***functional procedures*** (how geometric primitives and mechanical attributes are interconnected to create higher level motor skills), and ***behaviors*** (interrelationships between objects).

1.1 *Geometric Primitives*

The geometric description level is the foundation upon which each successive level is built. These descriptions comprise the bulk of information used to quantitatively classify a physically-based object. These geometric primitives are viewed as not only separate components, but also as the first level in a layered approach (Figure 2). The geometric descriptions of the functional model can be classified as:

1. One-dimensional point primitives
2. Two-dimensional surface primitives
   (e.g. polygons, patches)
3. Three-dimensional volume primitives
   (rigid or flexible)

One dimensional points and lines would describe natural phenomena that are processes or composed of discrete elements (e.g. clouds, water). Linear elements in our world (e.g. hair, string) can be represented as one-dimensional points linked together.

Two-dimensional primitives would be composed of two-dimensional planar surfaces connected together (e.g. paper, skin). Choices of linkages would determine the characteristic range of movement for an object. Hollow forms could be constructed from surfaces that are connected back on to themselves at their open edges (i.e. a flat surface curled into a cylinder shape).

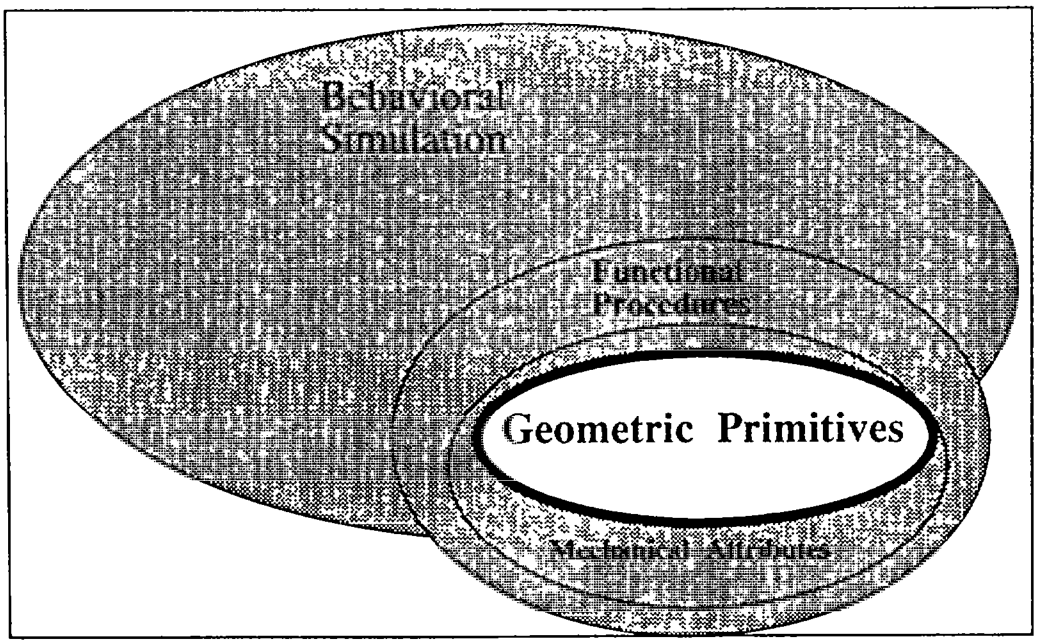

Figure 2, Geometric Primitives.
Structural description of an object or process.

Three-dimensional primitives (rigid or deformable). These would be composed of rigid (e.g. vases, bones, rocks) or deformable (e.g. muscles, jello) three-dimensional forms. The linking of these forms results in articulated structures which could be conceived of as a distinct sub-class within the domain of three-dimensional structures.

Deformable surfaces and structures include realistic forms such as strings (1D), paper (2D), jello (3D), and any subsequent combination. A form like a face can be constructed in layers accounting for the underlying rigid or semi-rigid structures underneath (bone [skull] then mass [fat, muscle] and then surface [skin]). It is from a combination of geometric primitives that articulated structures are composed. Articulated structures consist of rigid segments linked together (e.g. figures, animals, insects, trees).

Objects can be structurally defined not only through these primitives but also from other objects themselves. One object can be "part of" or a "movable part of" another.[9] For example, a piston (i.e. movable-part-of) is part of a motor (i.e. part-of) which is part of a motorcycle. The functional model proposed here is based on the assumption that geometric primitives are defined in a coordinate system, and the coordinate position of the object and its components are known locally or globally. This classification of one-dimensional, two-dimensional, and three-dimensional primitives constitute the foundation upon which mechanical attributes can be bound. Within the computer environment there is no such thing as a flexible object until mechanical attributes are associated with the object.

1.2 *Mechanical Attributes*

The mobile character of an object or agent (i.e. an object that can initiate action) is defined by its ***mechanical*** attributes. It is these mechanical qualities that permit internal and external movement. Mechanical attributes associated with an object includes joint linkages, mass, velocity, acceleration, deformation, force, torque, and surface area as a function of damping and collision detection. An object's characteristic movement is defined by these attributes. It is the combination of geometric primitives and mechanical attributes that permit a self-scripting or automatic simulation to proceed (Figure 3). The animator can affect modifications in the animation by changing mechanical attributes at this local level. A well-defined set of relationships or dynamic attributes will permit the logical assignment of mechanical attributes to the different geometries. These properties can be depicted as separate attributes:

Joint Linkages (connections between primitives) would be constrained to simulate a specified range of movement. Linkages in this model may differ from an artist's subjective concept of linkages. A linkage here determines the characteristic range of movement for the geometric primitives. Realistic joints serve as a reference for linkages. Such a listing would include: ball-and-socket joints, hinge joints, pivot joints, prismatic joints, gliding joints, condyloid, and saddle joints.[10]

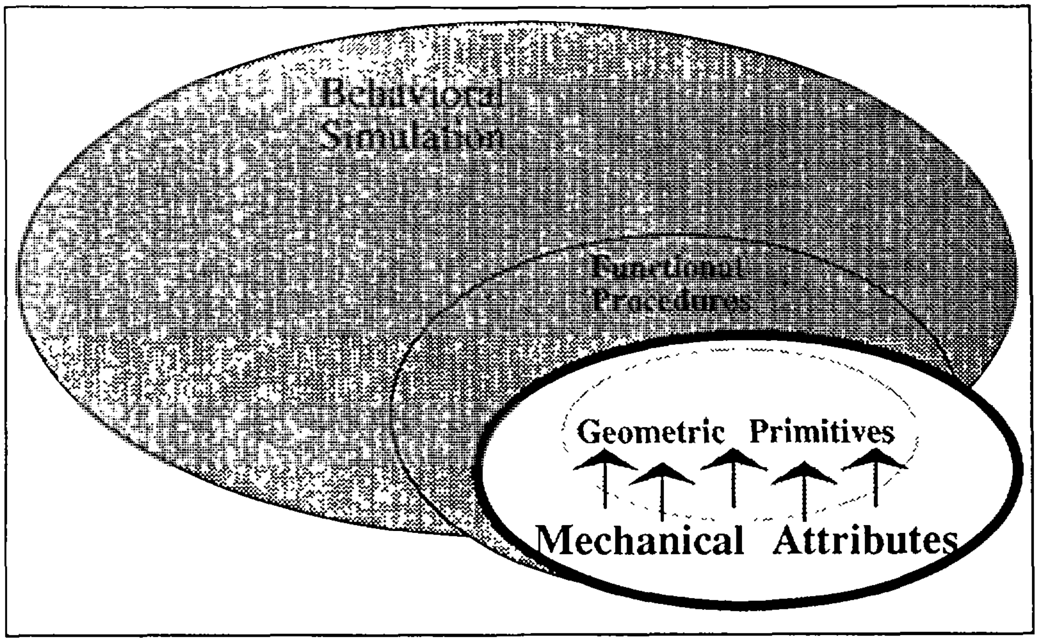

Figure 3, Mechanical Attributes.
The mobile character of an object is defined by these attributes.

Applications of such linkages can be at the micro (i.e. polygons) as well as the macro (i.e. objects) level. Specific linkages would be defined by range and type of constraints assigned to linkages. Under-constrained linkages would lend itself to proximity attachments; this would include rubberband type (i.e muscle to skin) linkages.

Mass attributes represent matter or weight at a point. This is the element that responds to environmental forces. It is responsible for resistance to changes in motion (i.e. inertia). This attribute is the basis upon which most other mechanical attributes interact. This primitive can be set globally for the object or locally at specific vertices on the object.

Velocity attributes contain the initial state of velocity an object has. Velocity is defined as the rate of change from one position to another position. In the real world all objects have a velocity relative to the environment, even if that is a zero velocity. All "states" are important. They represent the initial or current velocity which can be effected by a change in velocity.

Acceleration attributes would be the rate of change in velocity, the change from one velocity (i.e. zero, no movement) to another velocity (i.e. 5 MPH). This primitive would initiate changes in motion as the result of forces and masses interacting. Barr advocates including "impulse" attributes (linear and angular) to account for the initial change when an object begins to overcome inertia.[11] At that point it takes a greater force to get it going than to maintain velocity. The "impulse" attribute would be used to account for that difference.

Deformation attributes define the elasticity or stiffness of an object. This primitive contains information about an object's relative flexibility. This would be in the context of resistance to being pulled apart and/or pushed together. A necessary variable in this attribute would be whether the object kinetically absorbs the external force and changes shape or whether the force is released in the form of a reactive motion. For example, an aluminum can will deform in proportion to the magnitude of a force exerted on it while, on the other hand, a rubber ball will only momentarily deform before kinetically releasing the absorbed force in an observable reaction. The ball retains its original shape after a force, while the aluminum can deforms from its original shape.

Force attributes simulate external forces such as gravity and wind, or as internal forces resulting from muscle exertion. For example, gravity effects a downward force on the mass attributes. These forces are used by the dynamic motion procedures later.

Torque attributes would contain the magnitude of a force being applied in a joint to result in a desired rotation of an appendage at the joint. Again, this would be used by procedures at a higher level for dynamic motion simulation.

Surface-area attributes are a requisite for damping and collision detection. Damping is the motion of an object as the result of contact forces propagated by the surrounding fluid (i.e. wind or water). Damping can be computed through a formula that relates surface-area orientation to velocity vectors. Haumann has employed an effective ad-hoc technique to simulate damping through the use of a hinge joint at the polygonal level.[12]

1.3 *Functional Procedures*

For an object or agent with many links, it is desirable to be able to combine "geometric primitives" and "mechanical attributes" with "motion procedures" (e.g. dynamics, kinematics) into ***functional procedures*** which are necessary to effect a particular set of motions. Functional procedures (Figure 4) illustrate the combining of geometric primitives and mechanical attributes into functional procedures. Functional procedures permit the animator to create motor skills.[13] A prototypical physical object, for example, might obey some subset of the laws of Newtonian mechanics which can be assigned at the functional level. Articulated figures can build a repertoire of behaviors from these functions, such as walking and grasping.[14]

Movements that would be repeated frequently in an animation would be assembled into a reference library of motor skills. For example, the grasping movement of the hand can be factored into a functional procedure. From the known joint-angle rotations and hand movements functional primitive can be assembled

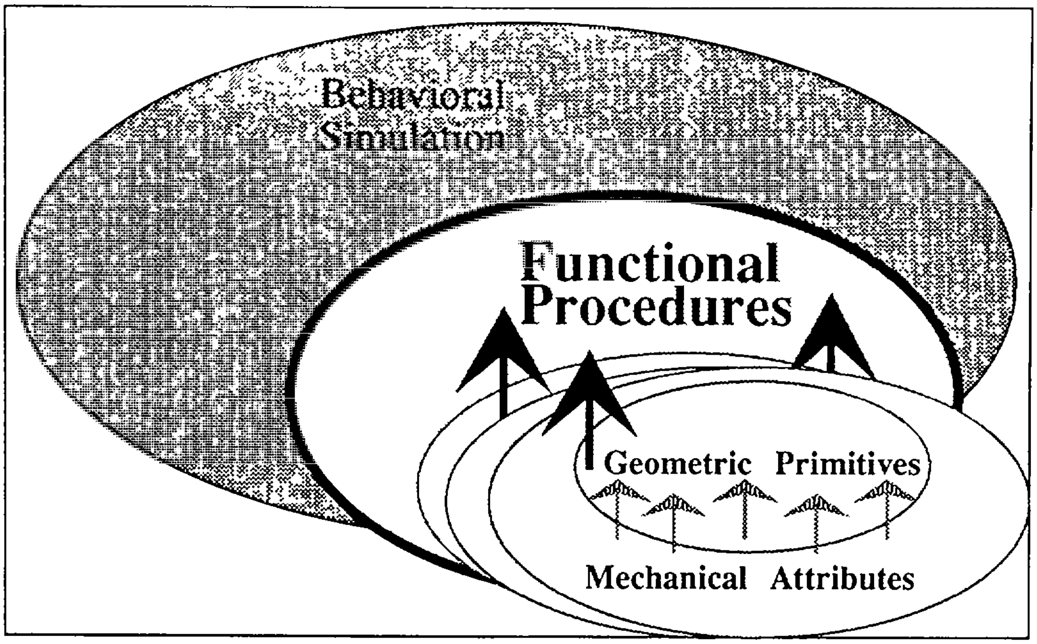

Figure 4, Functional Procedures permit the animator to combine geometric primitives and mechanical attributes to create higher level motor skills.

for "grasping". Once defined, the lower level details do not need to be redefined again later. This permits the animator "to direct" (e.g. target location, fast or slow, hard or soft)[15] the motion.

A functional procedure, like grasping, would be constructed as a kinematic or dynamic motion. It is at this functional level that the actual specification of mechanical attributes to specific geometric primitives would be assigned in conjunction with the desired type of motion (i.e. kinematic, dynamic). How these elements are hooked together directly affects the resultant motion of the objects. Haumann has suggested two useful levels: (a) "a coarse level for external constraints - for example: a complex object is related to the air by drag and to the ground by both gravity and contact," and (b) "a fine level for internal constraints - mass elements interconnected by spring and hinge elements to maintain internal object coherence."[16]

Functions can range from the simple specification of gravity to the complex motion of a scripted sequence (i.e. a dog getting a newspaper). Most important, these functional procedures may be nested together resulting in meta-behaviors.

1.4 Behavioral Simulation

The next level is ***behavior*** which is the result of many complex factors and interactions. Though physical motion can be simulated through Newtonian dynamics, *behavior* is more than the numerical solutions output from dynamic equations of motion. The difficulty in creating intelligent behavioral motion is that given some desired behavior (or property of behavior), we must find the forces which will produce it.[17]

The interrelationships between objects contain behavioral elements such as [a] programmable behaviors, [b] properties (mechanical, logical, social), and [c] local memory (event history, current state). The range of behaviors can be defined by a library of functional procedures and their interconnections. Hierarchical motion ***behaviors*** can be conceived of as several functional procedures combined to form a more complex functional procedure. This ***functions within a function*** concept permits the simulation of low-level behavior (Figure 5). Nevertheless, behaviors are also more than nested functions. True behavioral situations require objects to have local or global knowledge of their environments. That is, objects need the capacity to obtain information from their environment and also from other objects. If we wish to utilize goal-directed characters capable of achieving "non-trivial tasks then the character must take into account the geometry and mechanics of physical

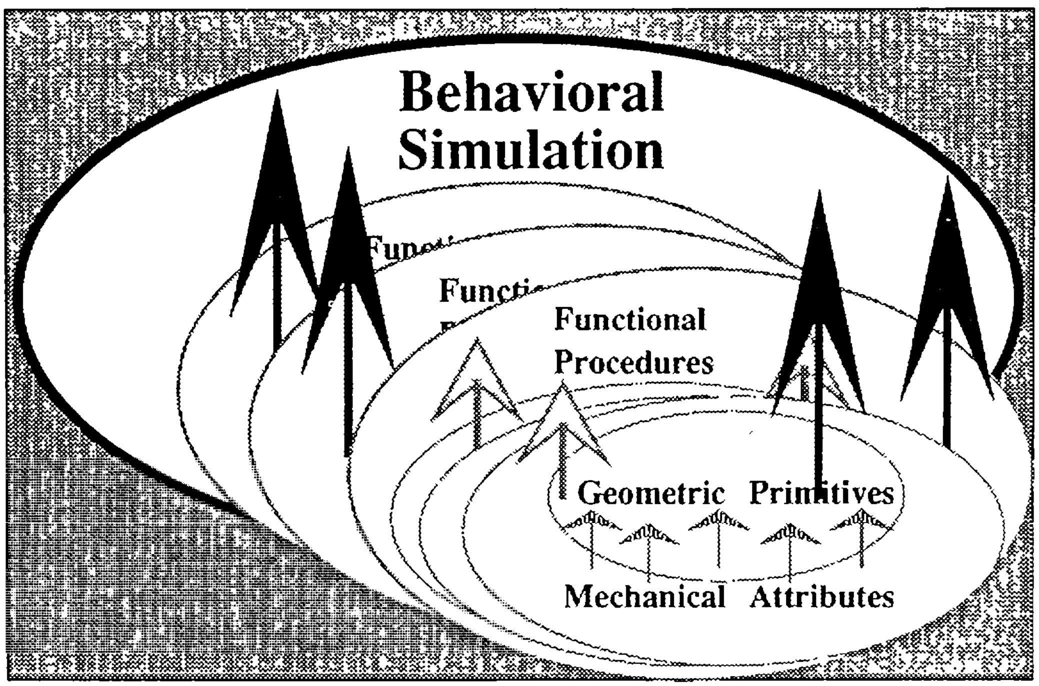

Figure 5, Behavioral Simulation.
The simulation of low-level behavior can be defined by the interconnections of a variety of functions. A *functions within a function* concept.

environments."[18]

Behaviors are not inherently locked to specific structures in this functional model. This ability to arbitrarily interconnect primitives leads to creative associations.

> While humans are in some sense active agents, they also obey the laws of Newtonian mechanics; a person falls just like a rock when pushed off a cliff. On the other hand, an animator may want the chairs and tables to dance around the room when the villain leaves. It should be easy to ascribe such behaviors to otherwise inanimate objects.[19]

Such creative associations will require motor problem solving within the animation system. In order to do simple motor problem solving, it will be necessary to embed common sense "knowledge" in object descriptions. That is, we want to be able to encode such default knowledge as one usually leaves a room by finding and opening the door. From our surrounding environment we have absorbed knowledge of naive physics, common sense, and mechanisms that are built on very non-conscious movements. Not only does this knowledge need to be accessible for use, but it also needs the option of being overridden once implemented so a character can leave by the window. This cognizance of the environment leads to intelligent motion behavior such as navigating through cluttered environments.[20]

Flexibility of the Model

To be able to weave all these primitives into a cohesive operational system will require a responsive interface. Interface issues of usability, flexibility, extensibility, and habitability as each relates to this functional model need to be implemented.[21] Of primary consideration is flexibility for the animator.

The goal of "flexibility" is to have the necessary constraints put on the system, not on the animator. A system should not force a way of working on the animator. Though a system should not impose physical laws on the animator, it should have them available when needed. As Gomez pointed out, "although Wily Coyote falls in a fashion that may be related to $d = \frac{1}{2}at^2$, it usually does not happen until he has been walking on air for a few seconds."[22]

The need to access different control levels stems from the inability of just one control mode (guiding, procedural, task) to provide the animator with complete yet reasonable control. Reynolds believes,

> ...in practice, most real animation is a combination of various techniques--certain characters may be created via behavioral simulation, while others in the same scene might be fully prescripted.[23]

The current prevalent ***guiding***[24] mode provides excellent refinement of explicit details but is too unweildy for controlling complex motion. Heavy reliance on this explicit level results in discontinuities in the motion. It is within this guiding mode that explicit geometric structures and mechanical attributes would be assigned.

The specification of functional procedures is located at the next level within the ***procedural*** mode.

The following four procedural methods of motion control are fundamental: ***forward kinematics, forward dynamics, inverse kinematics,*** and ***inverse dynamics. Forward kinematics*** permits the animator to manipulate an object or articulated figure by transformations in coordinate space. ***Forward dynamics*** also permits explicit placement but by means of forces and torques. ***Inverse kinematics*** and ***inverse dynamics*** permit the input of the position and orientation of a target location. From this transformation information the intermediate positions or torques and forces (necessary to reach the desired position or orientation) are computed. This "inverse" procedure automatically resolves the motion specifications needed. These procedures should be viewed as operating in a pipeline, with different motion procedures interacting with each other (Figure 6, Motion Pipeline and its Modules).

The pipeline organizes motion specification levels as modules. By linking the different modules together through a feedback control loop the artist has access to the different specification levels (guiding, procedural, task) when needed. These different levels permit artists to interject their desires either implicitly, explicitly, or algorithmically.

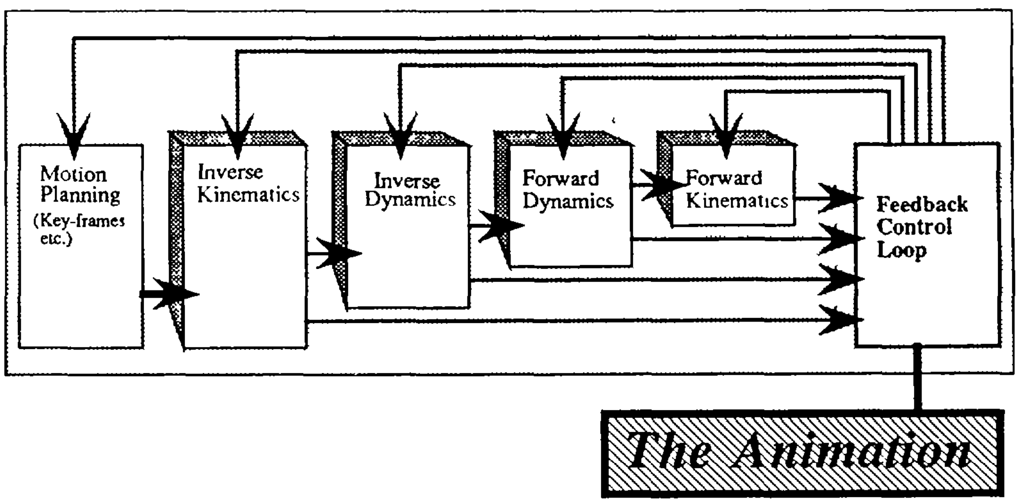

Figure 6, Motion Pipeline and its Modules.
Motion procedures are organized as operating in a pipeline,
with different motion procedures interacting with each other.

General motion planning schemes such as gait specification and path-planning will constitute high-level, implicit control. If "directed" or predictable control is desired the animator may choose the appropriate motion control module. For example, the inverse kinematics module permits the intuitive specification of objects by constraints. Kinematic constraints may be assigned in several ways. The most obvious is the pre-specification of position. Constraints may also come into effect when some inequality is satisfied, such as when one object attempts to occupy the same global position already occupied by another object. Constraints also "... may be invoked by a behavior based on current criteria in the system, (e.g., a ball stays in the hand after being caught until thrown)."[25] The inverse dynamic module can take the data from inverse kinematic specifications and determine force magnitudes that result from dynamic analysis. These forces can in turn be supplied to the forward dynamics module, which in turn can output rotational and translational values to the forward kinematic module. The feedback control loop provides a method of linking modules together: whether it be a straight sequence, an individual preference different modules, or repetitive loops of single or mixed modules. This ability to mix different modules can permit a keyframing animation system to be connected within a dynamic system.

The *Feedback Control Loop (Figure 7)* is the mechanism which provides for how these modules can be linked together and controlled. This process determines how closely the results may fulfill the artist's expectations. This control mechanism may be operated in several ways: [a] through explicit manipulation by the artist, [b] by the coupling of modules as procedures, [c] through the implicit goal-oriented direction, or [d] through predefined aesthetic-interpretative conventions in conjunction with evaluative criteria. In the feedback control loop we cannot only implement the explicit guiding control needed for fine tuning but also implement aesthetic controls of a higher order.

The significance of this loop is that animators are not handed one module but a collection of modules and possible connections from which to tailor the motion simulation to their vision. Rather than being confined to the specification of parameter values artists can now construct their own aesthetic algorithm from this model. It is the up to the artist as to select which motion generating modules are to activated and in what order.

It is anticipated that control will be initially focused at the guiding and procedural levels. It is here that the animator will specifically alter individual values or link (e.g. sequentially, intermixed, repeatedly) the different motion modules. The artist in the role of selective agent initiates action, views the results, and either accepts or rejects the outcome with the option to continue the process. The flexibility to interact on different levels with different modules provides a base for the concept of "browsers" as an interactive, procedural *what if?* tool.

The notion of *browsers* as implemented in Smalltalk (Tesler, 1981) or Loops (Stefik, 1983) suggests a powerful method for attaching guiding controls to motor skills. Suppose I have on my RGB monitor a shaded display of a human

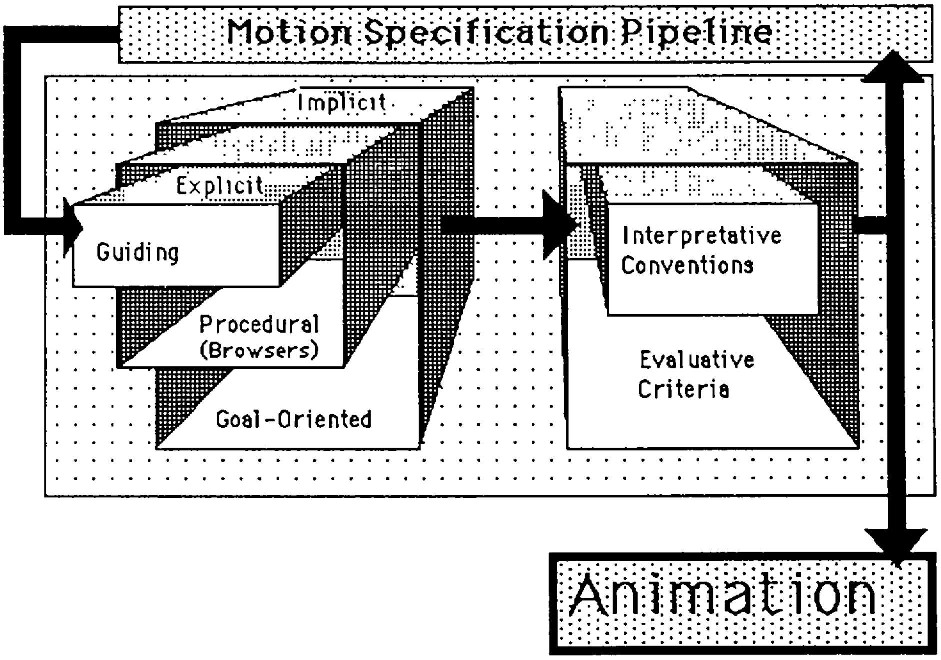

Figure 7, Feedback Control Loop.
This loop determines not only how modules can be linked together
but also how closely the results may fulfill the artist's goals.

character. On my terminal screen is a representation of the structure of the character and its skills. Now suppose I trace a curve on the graphics tablet. If I specify that that curve represents a particular joint rotation, - i.e., I point to the node for the little finger on my terminal, I should immediately see on the display the little finger of my character wiggling. Suppose now I point to the node for "grasping with the left hand" - I should see the figure's left hand open and close with the velocity I have specified. Lastly, if I pick the node labeled "walk", the figure should begin to walk across the screen, and this time, the curve I have drawn could determine, say, the speed of the gait.[26]

It is likely that "...the easiest way to specify a motion might be to specify the goal rather than 'how' to achieve the goal."[27] Such a *goal-oriented* mode is appropriate for the rough sketching out of an animation idea, or when only higher level control is needed. This mode gives implicit control over complex motions by trading off explicit command of the details. This goal-oriented mode is composed of the previous two - guiding and procedural.

AESTHETIC CRITERIA

A system derived from the functional model of this paper should not be limited to handling only visual complexities but should also be capable of handling "creative" complexities. That is, to simulate aspects of the creative process about which the artist already has some notion. These aspects address the concern that there is no new art form if the artist only automates the current animation process.

This aesthetic strategy could take the form of interpretative conventions and evaluative criteria.[28] Gips and Stiny have looked at the creative process as one in which external relationships and internal coherence can be codified into aesthetic algorithms.[29] These strategies permit formalized aesthetic viewpoints to be used to select and link motion modules according to predefined criteria. One of the first bodies of aesthetic information likely for this type of integration would be "Principles of Animation" (e.g. squash and stretch, anticipation, etc.)

Animation Principles - Continuity

Continuity in an animation can be achieved through the application of the known successful techniques (e.g. "Principles of Animation" from the book *The Illusion of Life* by Thomas and Johnson). A system implemented from the functional model described in this paper will be successful in direct relation to how its elements are applied. Film is not reality; it's a visual facsimile in need of creative devices to fill in the discontinuities inherent in the medium.

Simulating the ***squash-and-stretch*** principle of animation in computer animation can be accomplished primarily through techniques of "surface deformation" and secondarily through "motion blur" techniques. Motion blur alleviates the disturbing effects of temporal strobing. Temporal strobing is the disruption of the sequential perception of an image as it moves. Because there is no blurring effect the sequential position of an object becomes spaced far apart. This problem does not exist in live-action film because while the shutter is open the object's motion is recorded as a "smear" across the frame. This smearing contributes to the communication of continuity and in its own way contributes to the perception of ***squash-and-stretch.***

Elastic behavior can be built into the deformation of a form as a relationship between the kinematic and dynamic attributes of the object. This can result in ***squash-and-stretch, follow-through,*** and ***overlapping-action,*** and ***exaggeration*** principles being generated automatically .[30] (Figure 8) Chadwick and Parent[31] have suggested that prismatic joints, functioning as springs or shock absorbers can be used to form the foundation for exaggerated squash and stretch where needed.

Figure 8, Still image from the animation "Balloon Guy" by Chris Wedge. This animation utilized dynamic simulation software by Dave Haumann.

The principle of ***anticipation*** can be viewed as the anatomical provision for an action. It is a counter balance to the action impulse; the body stance that permits the action to be launched. The principle of ***follow-through*** and ***overlapping-action*** would be the natural dynamic consequence of an action; as would ***Slow in and slow out*** which deals with the spacing of the in-between drawings between the extreme poses. "Cel animators felt that one of the most objectionable traits of early computer animation motion was its lack of easing."[32] In most 3D key-frame computer animation systems the in-betweening is done automatically using spline interpolation.

Another principle of animation, ***secondary action*** is the reaction that results from an action. Secondary action can be accomplished through a collision detection mechanism or behavioral simulation. As an object collides or interacts with another, a force is transmitted which results in movement being propagated through the scene.

Appeal (the attraction or aesthetic quality of the work) might be considered one of the strongest points of this model. In animation, awkward, inconsistent, jerky, or unnatural motion results in a breakdown of the continuity and a lost of that illusion of reality created by the film medium. The use of simulation techniques results in a more fluid control of the motion continuity.

Whether the animator uses a key-frame method or a simulation approach, it will be necessary to organize the components in some responsive or interactive manner. Lasseter[33] points out that in working with a complex character, creating one complete pose at a time (all characteristics defined together) would make the in-between frames too unpredictable. Unexpected changes would materialize between pose extremes requiring numerous revisions of in-between frames.

In the context of hierarchical modeling there is a much better approach which works "layer by layer" down the hierarchy. Lasseter describes the process:

> Instead of animating one complete pose to another, one transformation is animated at a time, starting with the trunk of the hierarchical tree structure, working transformation by transformation down the branches to the end. Fewer extremes are used. Not all translates, rotates, and scales have extremes on the same frames; some have many extremes and others very few. With fewer extremes, the importance of the in-between frames increases.[34]

Chadwick makes the point that

> In order to effectively fine tune each degree of freedom precisely, each parameter is worked individually for a sequence of motion. Parameters are added and layered to build the desired motion. This effectively allows the user to isolate parameters which require fine-tuned adjustment.[35]

Organizing the parameters (e.g. translation, rotation, dynamics etc.) into a hierarchical system and having the animation proceed "layer by layer" down the hierarchy should prove to be a useful and likely paradigm in computer animation.

An animation system based on this functional model faces the dilemma of how much should be ready-made for the animator and how much should be constructed by the artist in the system itself. The animator is faced with a tradeoff of powerful options against efficiency. Ready-made procedures would contribute greatly to ease of control. However, these ready-made procedures will unintentionally guide an animator to sets of preconceived forms reminiscent of traditional art work. Using a ready-made system would negate the primary artistic use of a medium - to discover, create, and produce original imagery.

If the system relies heavily on the artist to specify a large quantity of parameters, some of which have non-intuitive values, then the motion may be very hard to control. Such a cumbersome situation could easily materialize if too many options are integrated in the system.

Requirements upon the Artist

This model introduces technical levels of complexity generally not found in recognized contemporary artistic methods. This will require a new type of artist, an interdisciplinary artist. Such an artist must be equipped to deal with the technical as well as the aesthetic. Csuri[36] states that it will require "knowledge, skill, perseverance, ingenuity and understanding backed by a sense of order, thought, purpose, and insight" on the part of this new artist. Artists-users who only allow themselves the depth of information that a list of instructions provides will not be "computer artists", just as the possession of a camera does not necessarily make one a photographer.

These new computer artists must be able to traverse the new paths being opened to them with the computer's assistance and to be able to exploit the expressive aspects of this model which emerge from logical processes and choices one makes in that process. In this regard, the medium can only be mastered to the degree that the artist has knowledge of the system. This is exemplified by Miller's[37] definition between "wiggle" and "wobble". These qualitative movements are readily known by artists but describing the difference in quantitative terms to the computer can be difficult - if not impossible - if artists do not understand the difference themselves. This is demonstrated by the following quantitative definitions:

wobble - resonant oscillation in response to external forces.

wiggle - deformation of shape due to internal forces.

The artist must understand the difference if they are to simulate the subtle difference that can exist between movements such as "wiggle" and "wobble".

The implication of this model is that the impact of simulation techniques as animation will be in the release of the animator's energy from the physical act of drawing, and re-focused into designing and directing. As artists aspire to create new original works of art, computer simulation will help them break new ground previously barred by constraints of time and complexity. It is the desire of this researcher that this model will provide artists desiring to explore the medium of computer simulation with a heuristic guide. This model has the potential of providing insights into creative possibilities that have yet to be conceived.

ACKNOWLEDGEMENTS

Thanks to Dave Haumann, Brian Guenther, Michael Girard, and John Chadwick for their considerable assistance and hours of feedback that helped clarify the problems faced in this endeavor. Thanks to Charles Csuri and Tom Linehan for the opportunity and stimulating environment at the Advanced Computing Center for Art and Design at the Ohio State University. Gratitude is expressed to all the special people that have made up the Advanced Computing Center for the Arts & Design team.

REFERENCES

1. David R. Haumann (1987). "Modeling the Physical Behavior of Flexible Objects," *SIGGRAPH '87 Tutorial Notes No. 17, Topics in Physically-Based Modeling.*

2. Alan H. Barr (1987). "Introduction to Physically-Based Modeling," *Tutorial Notes No. 17, SIGGRAPH '87, Topic in Physically-Based Modeling,*

3. Wilhelms, J. P. (1985). *Graphical Simulation of the Motion of Articulated Bodies such as Humans and Robots, with Special Emphasis on the use of Dynamic Analysis.* Unpublished doctoral dissertation, University of California, Berkeley.

4. David Zeltzer (1985). "Towards an integrated view of 3-D computer animation," *The Visual Computer, 1,* (4).

5. N. Magnenat-Thalmann & D. Thalmann (1985). *Computer Animation : Theory and Practice.* Tokyo: Springer-Verlag.

6. M. Girard (1987). "Interactive Design of 3D Computer-Animated Legged Animal Motion." *IEEE Computer Graphics and Applications, 7,* (6), 39-51.

7. Paul M. Issacs & M. F. Cohen (1987). "Controlling Dynamic Simulations with Kinematic Constraints, Behavior Functions and Inverse Dynamics," *ACM SIGGRAPH '87 Conference Proceedings, Computer Graphics, 21,* (3), p. 215.

8. Jack Burnham (1978). "System Esthetics," In, R. Kostelanetz (Ed.), *Esthetics Contemporary,* New York: Promethus Books, p. 160.

9. Zeltzer, *op. cit.,* p. 254.

10. Wilhelms, *op. cit.,* p. 30.

11. Barr, *op. cit.,* p. 2.

12. Haumann, *op. cit.*

13. Zeltzer, *op. cit.,* p. 252.

14. Ibid., p. 253.

15. Ibid.

16. Haumann, *op. cit.,* p. 5.

17. Michael Girard (1988). *An Overview of Motion Planning Problems in Robotics and Artificial Intelligence.* Unpublished manuscript, p. 1.

18. Ibid.

19. Zeltzer, *op. cit.,* p. 253.

20. Girard. *op. cit.,* p. 6.

21. Julian Gomez (1987). "Comments on Event Driven Computer Animation," *Notes for SIGGRAPH 87 Tutorial Computer Animation: 3D Motion Specification and Control,* pp. 77-90. Ibid., p. 127.

22. Ibid.

23. Craig Reynolds (1986). "Advanced Computer Animation", *ACM Siggraph '86, Tutorial Notes, Advanced Computer Animation,* p. 90.

24. Zeltzer,i *p. 253.*

25. Paul M. Issacs, & M. F. Cohen, (1987). "Controlling Dynamic Simulations with Kinematic Constraints, Behavior Functions and Inverse Dynamics," *ACM SIGGRAPH '87 Conference Proceedings, Computer Graphics, 21,* (3), p. 219.

26. Issac, *op. cit.,* p. 126.

27. Zeltzer, *op. cit.,* p. 257.

28. J. Gips and G. Stiny (1979). "An investigation of algorithmic aesthetics". In F. Malina (Ed.), Visual art, mathematics and computers: selections from the journal *Leonardo* New York: Pergamon Press, pp. 93-100.

29. Ibid.

30. John Lasseter (1987). "Principles of Traditional Animation Applied to 3D Computer Animation," *ACM SIGGRAPH '87 Conference Proceedings, 21,* (4), p. 41.

31. John Chadwick and Richard Parent (1988). "Critter Construction: Developing Characters for Computer Animation," *PIXIM Conference Proceedings, 1,* (1), p. 3.

32. Reynolds *op. cit.,* p. 90

33. Lasseter, *op. cit.*

34. Ibid.

35. Chadwick, *op. cit.* p. 3

36. Charles Csuri (1974). "Computer Graphics and Art," *Proceedings of the IEEE, 62,* (4), p. 514.

37. Presented at discussion of Galvin Miller's Paper, "The Motion Dynamics of Snakes and Worms," at the SIGGRAPH '88 Conference, Aug 4, 1988.