

[DETAILS](#)[COLUMNS](#)[CONTRIBUTIONS](#)

a

a

a

# RETROSPECTIVE

Vol.32 No.1 February 1998

ACM SIGGRAPH

## The State of Computer Animation

*David Sturman*  
**MaMaMedia**

For many of us in computer graphics and entertainment, the fascination is in the telling of stories with moving images. However the creation of convincing motion is not a trivial problem. After many years of continual technical developments, animation is still a labor intensive process requiring great skill and art.

Almost all computer animation today is done using keyframe systems evolved from the early 1980s. Only recently have advances such as full inverse kinematics, dynamics, flocking, automated walk cycles and 3D morphing made the leap from the academic to commercial sectors.

As a consequence, the big computer graphics production companies still rely on custom animation software, either alone or in concert with commercial packages, to fill in the gaps they leave in the animation toolkit.

Character animation is the hardest, requiring the most skill and receiving the most critical eye from the public. The difficulty of simulating natural living motion has led to a multitude of solutions, each attacking different aspects of the problem. Unfortunately, these solutions are developed in the laboratory, in isolation from everyday production tools, or within companies who understandably guard the techniques as trade secrets giving them a competitive edge. The commercial animation software producers are hard-pressed to keep up with the demands of the technology and generally lag five to 10 years behind the development community. They are trying hard to reverse this.

Another difficulty faced in computer animation is the global animation process itself. In commercial practice, an animation

requires a multitude of elements in dozens to hundreds of scenes and shots. Each scene may use several techniques of animation and rendering, involving numerous artists and animators. Coordinating these elements has always been one of the skills that make a good production company. However with the use of computers the process becomes more complex. As an animation passes through various stages of production, it may pass through various types of software. More often than not, these different software packages are not entirely compatible, imposing constraints on the production pipeline. There is almost no commercial software that addresses the problems of large CG animation projects either on the integration side or CG project management side. Most CG production companies have a staff of programmers to deal with these problems, and some of the older production companies rely on in-house software almost exclusively.

The main problem for commercial software producers has been how to integrate a diversity of animation methods into a single animation structure, or conversely, how to create an animation system that will accommodate a grab-bag of animation methods. 3D Studio approached the problem by providing an easy pathway for plug-ins. Hundreds of third-party developers have added diverse capabilities to the basic keyframe system including dynamics, facial animation, gait control and motion capture. Softimage and Alias|Wavefront have recently completely redesigned their systems to facilitate plug-ins and custom application development, as well as incorporating new animation methods in each release.

However, the problem is far from solved. Keyframe animation, although aided by many improvements, remains a time-consuming process. Recent advances in motion capture, performance animation and the incorporation of scripting into keyframe systems are beginning to break the dominance of keyframing and open up other avenues of creating animation. But these techniques are still new to production houses and their integration into the animation pipeline is far from complete.

## **How Did We Get Here And What Is Next?**

Perhaps one of the earliest pioneers of computer animation was Lee Harrison III. In the early 1960s, he experimented with animating figures using analog circuits and a cathode ray tube. Ahead of his time, he rigged up a body suit with potentiometers and created the first working motion capture rig, animating 3D

figures in real-time on his CRT screen. He made several short films with this system, called ANIMAC. This evolved into SCANIMATE which he commercialized to great success in 1969. SCANIMATE allowed interactive control (scaling, rotation, translation), recording and playback of video overlay elements to generate 2D animations and flying logos for television. Most of the 2D flying logos and graphics elements for television advertising in the 1970s were produced using SCANIMATE systems. In 1972 Harrison won an Emmy award for his technical achievements [25]. As computer graphics systems became more powerful in the 1980s, Harrison's analog systems began to be superseded by digital CG rendered keyframe animation, and now are no longer used in production.

The next widespread system was the GRaphics Symbiosis System (GRASS) developed by Tom DeFanti for his 1974 Ph.D. thesis. GRASS was a language for specifying 2D object animation and although not interactive, it was the first freely available system that could be mastered by the non-technical user. With GRASS, people could script scaling, translation, rotation and color changes of 2D objects over time. It quickly became a great hit with the artistic community who were experimenting with the new medium of CG. In 1978 it was updated to work in 3D with solid areas and volumes and ran on a Bally home computer. This version was called ZGRASS, and also was important in bringing computer graphics and animation to the artistic community on affordable computing platforms [6].

Also in 1974, Nestor Burtnyk and Marcelli Wein at the National Film Board of Canada developed an experimental computer animation system that allowed artists to animate 2D line drawings entered from a data tablet. Animation was performed by point-by-point interpolation of corresponding lines in a series of key frames. The system was used for 1974 classic short film *Hunger* whose graceful melding of lines from one figure to the next won it an Academy Award nomination.

The New York Institute of Technology Computer Graphics Lab (NYIT), then under the direction of Ed Catmull, extended this idea, producing a commercial animation system called TWEEN. As with the National Film Board system, TWEEN was a 2D system that allowed the animator to draw key frames, and the computer interpolated corresponding line segments between the keys. TWEEN automated the process of producing in-between frames, but still required the talents of a trained artist/animator for the keyframes. Although this method sped up the hand-animation process, animations produced this way had

an overly distinctive fluid look and the method was not widely adopted for commercial animation.

The first complete 3D animation systems were typically in-house tools developed for use in particular academic environments or production companies. They could be categorized into two types of systems, scripted or programmed systems, and interactive keyframe systems. The first type were exemplified by ANIMA-II [11], ASAS [23], and MIRA [16]. All three used a programming language to describe a time sequence of events and functions. When evaluated over time and a "snapshot" rendered at each animation frame, they produced the desired animation. ASAS is noteworthy since many of the CG sequences in the 1982 film TRON were animated with it. These systems were powerful in that almost anything could be done if it could be programmed, but limited in that programming skills were required to master them.

The keyframe systems were more amenable to animation artists. Based on the keyframe approach of traditional animation, these systems allowed the user to interactively position objects and figures in the scene, save these positions as keyframes and let the computer calculate the in-between frames to produce the final animation. GRAMPS [19] and BBOP [28] were examples of this type of system. Both relied on the real-time interactivity of the, then state-of-the-art, Evans & Sutherland Multi-Picture System, an excellent vector-graphics display system that worked from a display list allowing instantaneous updates of the on-screen graphics.

GRAMPS was developed for visualization of chemical structures although O'Donnell does give examples of how it could be used to animate a human figure. Ostensibly an interpreted script system, GRAMPS allowed script variables to be connected to dials for interactive manipulations.

### **An interesting aside:**

#### **2D and 3D animation.**

BBOP was developed at the New York Institute of Technology's Computer Graphics Lab (NYIT) by Garland Stern expressly for character animation and was used extensively by NYIT in six years of commercial production. In BBOP, animators could interactively control joint transformations in a 3D hierarchy, saving poses in keyframes which the computer could interpolate to produce smooth animation. The system was very responsive and easy to use, and conventionally trained animators produced some remarkably good animation with it. Examples include a CG football player for Monday Night

Football promotions, Rebecca Allen's CG dancer for Twyla Tharp's dance The Catherine Wheel, Susan Van Bearle's short film Dancers and numerous SIGGRAPH film show shorts featuring the digital characters User Friendly, Dot Matrix and User Abuser. These last three were some of the first CG characters to have expressive personalities that engaged the audience and brought CG to life. Much of this was due to an interactive keyframe system that gave the animator the control and freedom to manipulate the figures visually, in keeping with his training and experience.

Most modern commercial keyframe systems are based on the simple BBOP interactive keyframe approach to animation with added features that ease the animation process. At their core, they all have features of BBOP (some copied, some developed independently), including hierarchical skeleton structures, real-time interactive update of transformation values, interpolation of keyframes in channels so that different joints can have different keys in different frames, choice of interpolation functions such as linear, cubic, ease-in and ease-out, immediate playback and an interpolation editor.

In general, however, scripted systems still are best for repeated or easily describable movements, but require programming skills beyond the capabilities of most artists, especially as movements become more complex. Scripting expressive characters, for example, is extremely difficult, not to mention unnatural for an artist. Interactive keyframe systems are just the opposite. They allow artists to interact directly with the objects and figures within a familiar conceptual framework. But they become inefficient or tedious to use for mechanical or complex algorithmic motion. Because they are more easily used by artists, the interactive keyframe approach has won in the commercial software market. Curiously enough, as animators are becoming more sophisticated in their use of computer animation, scripting capabilities are beginning to reappear in keyframe systems. The newest version of Wavefront|Alias' MAYA animation system has a built-in scripting capability that allows animators to tie actions to events, define movement as functions of other movements, create macros and more.

Early 3D animation systems mostly dealt with simple forward kinematics of jointed bodies, however inverse kinematics can also be an important element in an animation toolkit. By moving just a hand or a foot, the animator can position an entire limb. Michael Girard built a sophisticated inverse kinematic animation system for his Ph.D. thesis [9] which was used for producing

very graceful human body movement in his 1989 film *Eurhythmy*. He later commercialized his system as a 3D Studio MAX plug-in, *Biped* (part of the *Character Studio* package), where legged locomotion such as walks, runs, jumps and skips can be animated by placing footprints. His inverse kinematic algorithms compute the motions of the figure that cause it to follow the footprints.

When *Softimage* was first released, it was the first commercial system to feature an inverse kinematics capability (although in a simplified form). That feature helped greatly in selling the new system. Now, almost all 3D animation systems have some form of inverse kinematic capabilities.

Dynamics is also an important tool for realistic animation. Jane Wilhelms was one of the first to demonstrate the use of dynamics to control an animated character [31]. Since then, James K. Hahn, David Baraff and Michael McKenna [12, 2, 17] have all described robust dynamics for computer animation. Yet it is only in the past few years that the major commercial systems are incorporating dynamics into their software. The problems they are facing are how to integrate dynamic controls, inverse kinematic controls, and forward kinematic controls within the same system, and presenting and resolving clearly the potentially conflicting constraints each puts on the animated elements.

Kinematics and dynamics deal with jointed skeletal structures. However, not all animation is skeletal. A face, for example, is a single surface with complex deformations. Fred Parke was the first to attack this problem [20] with a parametric facial model. Using parameters to describe key aspects of facial form, such as mouth shape, eye shape and cheek height, and then animating these parameters, he was able to simulate the motions of a human face as a single surface. The system was used by NYIT in a music video for the group *Kraftwerk*, but never commercialized.

Years later, Philippe Bergeron and Pierre Lachapelle digitized plaster models of several dozen expressions of a face, and created a system to interpolate between several of these target expressions at once for their 1985 short film *Tony de Peltrie* [3]. The result was a rubbery-faced character with a wide range of human expression. Rudimentary implementations of this technique of 3D object interpolation (or 3D target morphing) were incorporated into *Softimage* and *Alias|Wavefront* systems a few years ago, and are being improved for the latest versions of their software. 3D target morphing is also the basis of

Medialab's real-time character performance animation system.

Keith Waters developed an even more sophisticated facial animation system based on muscle activation influencing regions of the face model [30]. This system produces very realistic facial motion and can be controlled by high-level commands to the muscle groups. His methods have not been commercialized, but simpler versions are used in some optical facial motion capture systems.

There follow a whole host of bits and pieces to animate particular effects. Some of these have been integrated into commercial animation systems. Others are used exclusively by the companies that developed them, while still others have just seen proof of concept and await a plug-in or incorporation into a more complete system.

The most influential of these (and perhaps not really in the bits and pieces category) is Bill Reeve's particle systems [22]. Reeves developed a method of using controlled random streams of particles to simulate fire, grass, sparks, fluids and a whole host of other natural phenomena. First used in the movie *Star Trek II*, particle systems are easy to implement and quickly appeared in many amateur, academic and professional CG animations, most notably *Particle Dreams* in 1988 by Karl Sims. Commercial animation systems took a little longer to incorporate the technique into their established structures, but today everyone has it in some form or another.

Other animation techniques for specific effects in the literature include (but by no means are limited to) automated gaits (walking, running, jumping, etc.) [5, 13], flocking behaviours [24, 1], fluid flow [14], waves [7, 21], smoke [27], sand [15], flexible objects [29], snakes [18], cloth [29] and many more.

As was already mentioned, the most difficult animation is character animation, particularly human character animation. In a quest for more realistic motion, people have looked towards directly recording the motions of a human performer. Lee Harrison III in the 1960s was only the first of many to use this concept. In 1983 Ginsberg and Maxwell [8] presented an animation system using a series of flashing LEDs attached to a performer. A set of cameras triangulated the LEDs' positions, returning a set of 3D points in real time. The system was used soon after to animate a CG television host in Japan. However motion capture systems and graphics computers were just not fast enough then for the real-time demands of performance animation.

When they did begin to become fast enough, around 1988 with the introduction of the Silicon Graphics 4D workstations, deGraf/Wharman and Pacific Data Images both developed mechanical controllers (also known as waldos) to drive CG animated characters -- deGraf/Wharman for CG facial animation for a special SIGGRAPH presentation and for the film Robocop II, and PDI for a CG character for a Jim Henson television series and several other projects. For various reasons the technology and market were not ready and the systems were rarely exploited after their initial use.

Then, in the early 1990s, SimGraphics, Medialab (Paris) and Brad deGraf (with Colossal Pictures and later Protozoa) all independently developed systems that allowed live performers to control the actions of a CG character in real time. These systems allowed characters to be animated live, as well as for later rendering. The results, particularly with Medialab's system, are characters that have very lifelike and believable movements. Animation can be generated quickly by actors and puppeteers under the control of a director who has immediate feedback from the real-time version of the character. All three systems have survived their initial versions and applications, and continue to be successfully used in commercial projects.

At first, these systems existed on their own and were not integrated into other commercial CG systems. Animation done in a keyframe system could not easily be mixed with animation performed in a real-time system. As time has passed, both the real-time systems and the keyframe systems have evolved, and now many keyframe systems have provisions for real-time input and the real-time systems import and export keyframe animation curves.

Performance animation has become very popular recently and at the SIGGRAPH 97 trade show, no less than seven companies demonstrated performance animation systems.

Similar to performance animation, but without the real-time feedback, are motion capture systems. These are generally optical systems that use reflective markers on the human performer. During the performance, multiple cameras calculate the 3D positions of each marker, tracking it through space and time. An off-line process matches these markers to positions on a CG skeleton, duplicating the performed motion. Although there are problems with losing markers due to temporary occlusions and the animation matching process can be very labor-intensive, motion capture permits an accurate rendering of human body motion, particularly when trying to simulate the



motion of a particular performer as Digital Domain did with Michael Jackson's 1997 music video, Ghosts.

## **This Brings Us To The Present**

More and more production studios are combining techniques from keyframe animation systems, plug-ins, custom code, motion capture and performance animation in order to improve the quality and efficiency of their animation. As commercial systems become more open and provide plug-in capabilities, animation techniques that previously lived and died outside of the production domain will be integrated rapidly into the mainstream, augmenting the animation toolkit to the benefit of all. As systems become more open, it will be easier to transfer between one and another, easing the production pipeline. As the tools are standardized, companies can begin to develop software to manage the CG production process, making large CG productions all that much easier. For example, there are several 3D CG Saturday-morning cartoon series in production or pre-production following the success of Reboot and Beast Wars. The single hardest technical problem they will encounter is the efficiency of their CG production pipeline.

As hardware becomes faster and less expensive, animation stations are becoming affordable to even the smallest companies and to individuals. This, combined with improved tools for developing plug-ins, may give rise to artists developing and distributing their own animation techniques, opening the medium of computer animation with new images and styles. With enough independents in the mix, we may find a radical new way of doing animation that facilitates the process a thousand-fold.

The Internet provides another host of new opportunities for animation. Using VRML, both blitcom (LA) and Protozoa (SF), are already broadcasting episodes of computer animated cartoon series. With a greater variety of inexpensive animation tools, this capability will be available to almost anyone. Not long ago, Microsoft researchers released software to design avatars for one of their experimental virtual meeting places. A young teenage user became so adept at creating interesting characters that other users were paying him for custom avatars.

Advances in computer vision will also revolutionize the field. The [MIT Media Lab's Vision and Modeling Group](#) is developing powerful algorithms for understanding human body movement. Their work will allow future motion capture systems to operate

without the performer having to wear special sensing equipment. Extending this concept, one could imagine children acting out characters and scenes, directly creating animated videos for fun, school or to share with their friends.

As computers understand human motion, they will be better able to assist us in generating believable movement in synthetic characters, leading to higher-level specification of animation. As animated characters become more "intelligent" and autonomous, they can be used more and more like actors, with the director giving spoken language commands. Bruce Blumberg [4] describes a system in which autonomous characters can be directed. They have lives and behaviors of their own in the absence of direction, but when directed, they are compliant performers. This leads to synthetic performers that can improvise and add personality to directed performances. Karl Sims [26] and Radek Grzeszczuk [10] developed animated entities that evolve and learn from experience. From this one could imagine creating animated characters by evolving and training them to have certain behaviors.

There will always be animation that needs careful, artistic, hands-on attention, but as we make advancements, more and more "intelligent" characters will fill in the ranks of the painstakingly animated. This will not only open up computer animation to many who would do not have the current requisite technical skills, but free up skilled artists to concentrate on the more interesting aspects of expression through animation.

### Apology

It would be impossible in the space given to have presented all the different aspects, problems and solutions of computer animation. Instead I have tried to paint a broad picture of where we've been and where we are today. There are many contributions to the field that I have omitted or glossed over; they are nevertheless important and if I have missed an old favorite, I apologize for not being able to include it here.

**David Sturman** is Vice President Technology for MaMaMedia, a children's Internet company creating playful, activity-based

### References

1. Amkraut, Susan and Michael Girard. "Eurhythmy: Concept and Process," *Journal of Visualization and Computer Animation*, v.1, n.1, John Wiley & Sons, West Sussex, England, August 1990, pp. 15-17.

tools that build technological fluency .  
Formerly Director of Research and Development for Medialab, Paris, Sturman has a Ph.D. from the MIT Media Lab and has been active in the field of computer animation since 1983 when he was a member of the technical staff at the NYIT Computer Graphics Lab.

### David Sturman

MaMaMedia  
110 Greene Street  
Suite 805  
New York, NY 10012

### Web site

The copyright of articles and images printed remains with the author unless otherwise indicated.

2. Baraff, David. "Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies," *Computer Graphics, proceedings of SIGGRAPH 89*, ACM SIGGRAPH, New York, NY, pp. 223-232.
3. Bergeron, Philippe and Pierre Lachapelle. "Controlling Facial Expressions and Body Movements in the Computer-Generated Animated Short Tony De Peltrie ," *SIGGRAPH 85 Advanced Computer Animation seminar notes*, ACM, New York, 1985.
4. Blumberg, Bruce M. and Tinsley A. Galyean. "Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments," *Computer Graphics, proceedings of SIGGRAPH 95*, ACM SIGGRAPH, New York, NY, pp. 47-54.
5. Bruderlin, Armin and Thomas W. Calvert. "Goal-Directed, Dynamic Animation of Human Walking," *Computer Graphics, proceedings of SIGGRAPH 89 Proceedings*, ACM SIGGRAPH, New York, NY, pp. 233-242.
6. DeFanti, T., J. Fenton and N. Donato, "BASIC Zgrass -- A sophisticated graphics language for the Bally home computer," *Computer Graphics, proceedings of SIGGRAPH 78*, ACM SIGGRAPH, New York, NY, pp. 33-37.
7. Fournier, Alain and William T. Reeves. "A Simple Model of Ocean Waves," *Computer Graphics, proceedings of SIGGRAPH 86*, ACM SIGGRAPH, New York, NY, pp. 75-84.
8. Ginsberg, Carol M. and Delle Maxwell. "Graphical marionette," *Proceedings ACM SIGGRAPH/SIGART Workshop on Motion*, Toronto, Canada, April 1983, pp. 172-179.
9. Girard, Michael and Anthony A. Maciejewski. "Computational Modeling for the Computer Animation of Legged Figures," *Computer Graphics, proceedings of SIGGRAPH 85*, ACM SIGGRAPH, New York, NY, pp. 263-270.
10. Grzeszczuk, Radek and Demetri Terzopoulos. "Automated Learning of Muscle-Actuated Locomotion Through Control Abstraction," *Computer Graphics, proceedings of SIGGRAPH 95*, ACM SIGGRAPH, New York, NY, pp. 63-70.
11. Hackathorn, Ronald J. "Anima II: a 3-D Color Animation System," *Computer Graphics, proceedings of SIGGRAPH 77*, ACM SIGGRAPH, New York, NY, pp. 54-64.
12. Hahn, James K. "Realistic Animation of Rigid Bodies," *Computer Graphics, proceedings of SIGGRAPH 88*, ACM SIGGRAPH, New York, NY, pp. 299-308.

13. Hodgins, Jessica K., Wayne L. Wooten, David C. Brogan and James F. O'Brien, "Animating Human Athletics," *Computer Graphics, proceedings of SIGGRAPH 95*, ACM SIGGRAPH, New York, NY, pp. 71-78.
14. Kass, Michael and Gavin Miller. "Rapid, Stable Fluid Dynamics for Computer Graphics," *Computer Graphics, proceedings of SIGGRAPH 90*, ACM SIGGRAPH, New York, NY, pp. 49-57.
15. Li, Xin and J. Michael Moshell. "Modeling Soil: Realtime Dynamic Models for Soil Slippage and Manipulation," *Computer Graphics, proceedings of SIGGRAPH 93*, ACM SIGGRAPH, New York, NY, pp. 361-368.
16. Magnenat-Thalmann, N. and D. Thalmann. "The Use of High-Level 3-D Graphical Types in the Mira Animation System," *IEEE Computer Graphics and Applications*, v.3, 1983, pp.9-16.
17. McKenna, Michael and David Zeltzer. "Dynamic Simulation of Autonomous Legged Locomotion," *Computer Graphics, proceedings of SIGGRAPH 90*, ACM SIGGRAPH, New York, NY, pp. 29-38.
18. Miller, Gavin. "The Motion Dynamics of Snakes and Worms," *Computer Graphics, proceedings of SIGGRAPH 88*, ACM SIGGRAPH, New York, NY, pp. 169-178.
19. O'Donnell, T.J. and A. J. Olson. "GRAMPS -- A graphics language interpreter for real-time, interactive, three-dimensional picture editing and animation," *Computer Graphics, proceedings of SIGGRAPH 81*, ACM SIGGRAPH, New York, NY, pp. 133-142.
20. Parke, Frederic I. "Computer Generated Animation of Faces," *Proceedings ACM annual conference*, August 1972.
21. Peachey, Darwyn R. "Modeling Waves and Surf," *Computer Graphics, proceedings of SIGGRAPH 86*, ACM SIGGRAPH, New York, NY, pp. 65-74.
22. Reeves, W. T. "Particle Systems -- a Technique for Modeling a Class of Fuzzy Objects," *ACM Trans. Graphics*, v.2, April 1983, pp. 91-108.
23. Reynolds, C. W. "Computer Animation with Scripts and Actors," *Computer Graphics, proceedings of SIGGRAPH 82*, ACM SIGGRAPH, New York, NY, pp. 289-296.
24. Reynolds, Craig W. "Flocks, Herds, and Schools: {A} Distributed Behavioral Model," *Computer Graphics, proceedings of SIGGRAPH 87*, ACM SIGGRAPH, New York, NY, pp. 25-34.
25. Schier, Jeff. "Early Scan Processors-- ANIMAC/SCANIMATE," in *Pioneers of Electronic Art*, Exhibition catalog, Ars Electronica 1992, ed. David Dunn,

- Linz, Austria, 1992, pp. 94-95.
26. Sims, Karl. "Evolving Virtual Creatures," *Computer Graphics, proceedings of SIGGRAPH 94*, ACM SIGGRAPH, New York, NY, pp. 15-22.
  27. Stam, Jos and Eugene Fiume. "Turbulent Wind Fields for Gaseous Phenomena," *Computer Graphics, proceedings of SIGGRAPH 93*, ACM SIGGRAPH, New York, NY, pp. 369-376.
  28. Stern, Garland. "Bbop - a Program for 3-Dimensional Animation," *Nicograph '83*, December 1983, pp. 403-404.
  29. Terzopoulos, Demetri, John Platt, Alan Barr and Kurt Fleischer. "Elastically Deformable Models," *Computer Graphics, proceedings of SIGGRAPH 87*, ACM SIGGRAPH, New York, NY, pp. 205-214.
  30. Waters, Keith. "A Muscle Model for Animating Three-Dimensional Facial Expression," *Computer Graphics, proceedings of SIGGRAPH 87*, ACM SIGGRAPH, New York, NY, pp. 17-24.
  31. Wilhelms, J. and B. A. Barsky, "Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots," *Graphics Interface '85 Proceedings*, Montreal, Quebec, Canada, May 1985, pp. 97-104.