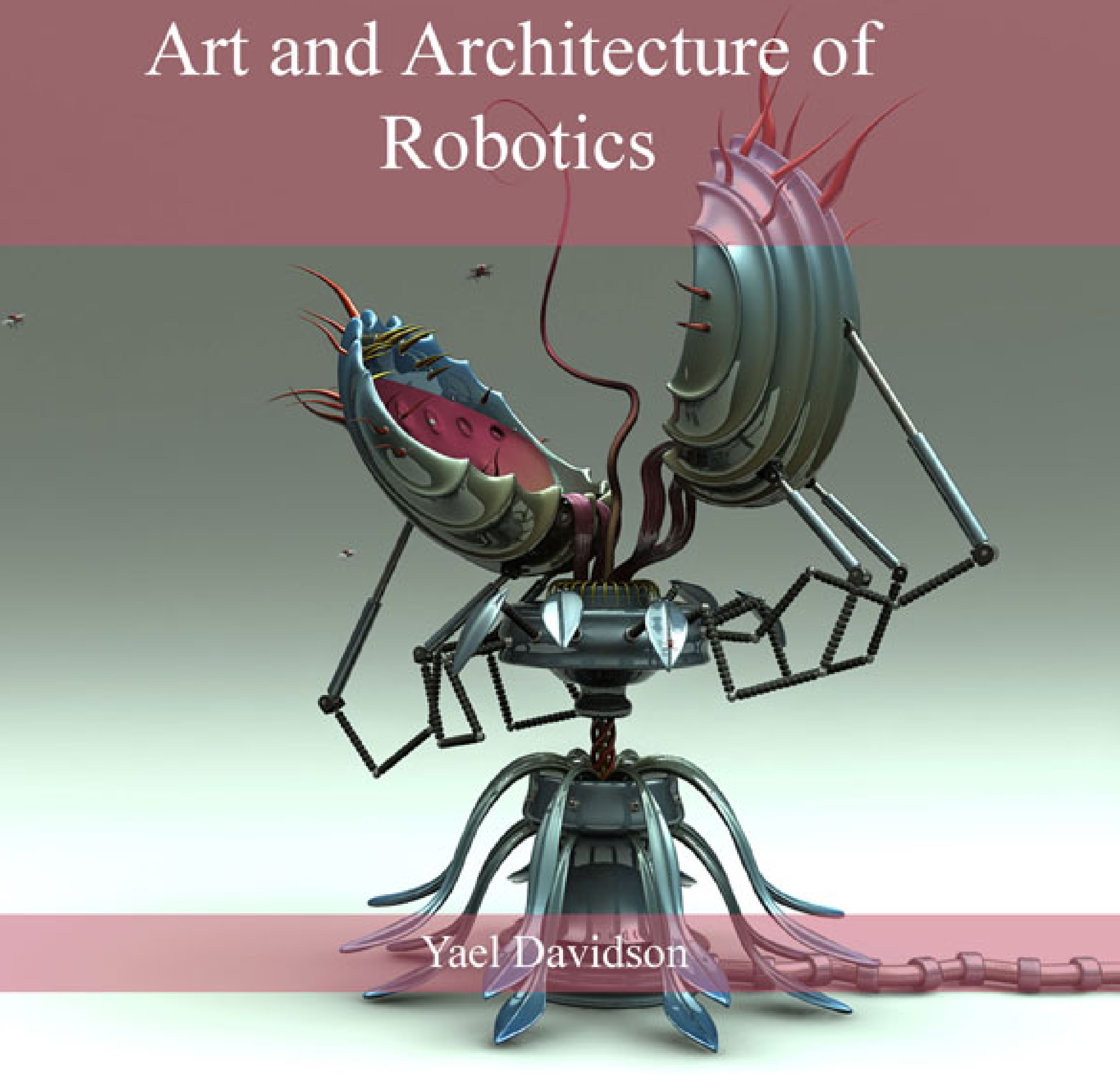
Art and Architecture of Robotics



Yael Davidson

First Edition, 2011

ISBN 978-93-81157-94-7

Published by:
The English Press
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Art of Robotics

Robotic art at Robogames 2008, San Francisco

Robotic art is a broad term that encompasses a variety of sub-types of art, all of which employ some form of robotic or automated technology.

Robotic installation art unifies Installation art and robotic technologies insofar as the works and installations often employ computers, sensors, actuators and programming which allow them to respond or evolve in relation to viewer interactions. In this kind of art and technology-based work the viewer is transformed from a passive viewer to an active participant. One significant way in which this work can differ from kinetic art is that it is usually non-programmatic in the sense that the future behavior of the sculpture or installation can be altered by input from either the artist or the participant.

History

The history and evolution of robotic art and theater is quite involved. Early progenitors start in Ancient China (Han Dynasty, c. third century B.C.), with the development of a mechanical orchestra, and other devices such as mechanical toys. These last included flying automatons, mechanized doves and fish, angels and dragons, and automated cup-bearers, all hydraulically-actuated for the amusement of Emperors by anonymous engineer-craftspeople. Several names have come down to us, however. Mo Ti and the artificer Yen Chin are said to have created automated chariots. By the time of the Sui Dynasty (6'th Century A.D.), a compendium was written called the Shai Shih t'u Ching, or 'Book of Hydraulic Excellencies'. There are reports that the T'ang Dynasty saw Chinese engineers building mechanical birds, otters that swallowed fish, and monks begging girls to sing.

Rome

In Ancient Rome in the time of Nero the great poet and novelist Petronius made a “doll that moved”, and around c. 85 A.D. there were the amazing writings and creations of Hero of Alexandria, who wrote "On Automatic Theaters, On Pneumatics, and on Mechanics", and is said to have built fully-automated theatrical set-pieces illustrating the labors of Hercules among other wonders.

Thirteenth century AD

In the 13th century AD Badi Al-Zaman'Isma'il Al-Razzaz Al-Jazari was a Muslim inventor who devoted himself to mechanical engineering. Like Hero, he experimented with water clock and other hydraulic mechanisms. Al-Jaziri’s life's work culminated in a book which he called “The Book of Knowledge of Ingenious Mechanical Devices,” completed in 1206 AD. This book is often known simply as “Automata.” In Europe in the 13th century Villard de Honnecourt is known to have built mechanical angels for the French court, and in the 15th century Johannes Muller built both a working mechanical Eagle and a Fly.

Middle Ages

In the Middle Ages in Europe, clockmakers built an astronomical Clock in Prague. On the hour, a skeleton with an hourglass in his hand rings a bell, and then a Turk draws his sword. Finally a series of animatronic figures move across the top of the clock.

Examples

- Leonardo da Vinci invented several theatrical automata including a lion which walked onstage and delivered flowers from its breast, and a soldier.

- A mechanical theater at the gardens of Hellbrun near Salzburg, Austria contained over 113 hydraulically operated figures.

- Magician Isaac Fawkes, in 1722, used a clock that "played a variety of tunes on the organ, flute and flangolet with birds whistling and singing." He also had a mechanism called the "Temple of the Arts," which featured mechanical musicians, ships and... ducks.

- Pierre Jaquet-Droz was a Swiss watchmaker who made some of the most amazing and sophisticated automotas ever seen, including The Writer (made of 6000 pieces), The Musician (2500 pieces) and The Draughtsman (2000 pieces). These devices are mechanical analog computers and form an essential link in the evolutionary chain of the development of robotic culture. They are still to be seen in working condition at the art and history museum in Neuchâtel, Switzerland. Automatic chess players, artists and other figures were made with increasing frequency in the 18th and 19th centuries.

Humanoid robots

As the industrial revolution grew a new sub-genre of literature addressing the anxieties of the age appeared. Many of these writings featured a mechanical humanoid as a central character. Some of these artificial men in literature included:

- *The Nightingale* – Andersen
- *Frankenstein* – Shelley
- *The Belltower* – Melville
- *The Artist of the Beautiful* – Hawthorne
- *Moxon's Master* – Bierce
- the 'Golem' of Jewish folklore.
- *The Wizard of Oz* – Baum

and in the early 20th century:

- *R.U.R.* – Capek
- *Runaround* – Asimov

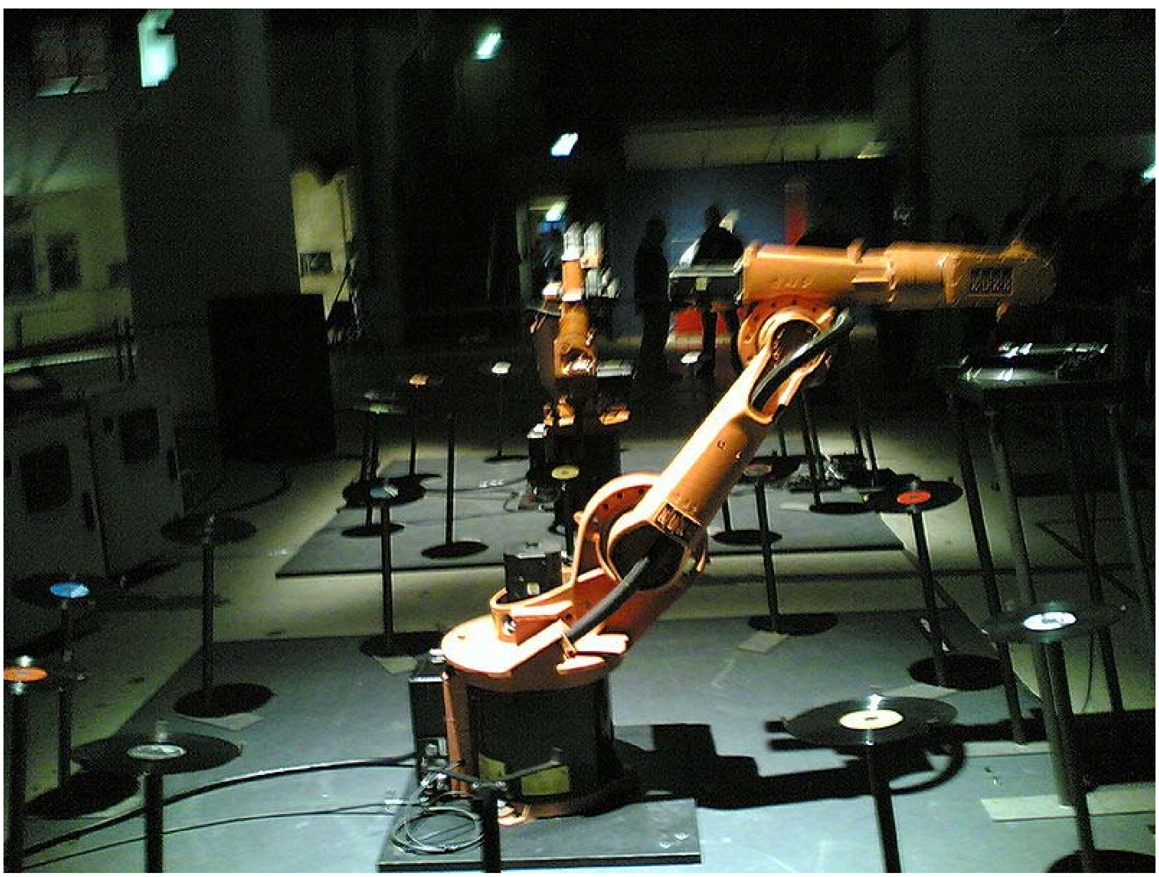
Engineering

Advances in engineering created new possibilities. In 1893 Prof. George Moore created 'The Steam Man', a steam-powered robot in New York City which reportedly pulled a wagon-load of musicians in a parade. Rumour has it that parts from this Steam Man appeared in junk shops around Manhattan a few years later.

The revolutionary work of Nikolai Tesla, is an example. In 1898 Tesla demonstrated a remote-controlled robotic submarine in Madison Square Garden. Tesla described this historic vehicle as having "a borrowed mind. When first shown... it created a sensation such as no other invention of mine has ever produced."

Robotics have now become a mode of expression for artists confronting fundamental issues and contradictions in our advanced industrial culture.

Performance Art



German artist group **robotlab** works with industrial KUKA robots in public spaces. It explores the relationship between man and machine by means of installations and performances. "juke bots" is an installation, in which two robot arms are creating sound by means of records.

Robotic performance art refers to the presentation of theatrical performances in which most, if not all, of the "action" is executed by robots rather than by people. An early robotic artist was Edward Ihnatowicz, who created The Senster (1969–71). It employed sound sensors and hydraulics, which reacted to visitors in the space. Shows of this sort are sometimes large and elaborate productions. The Swiss sculptor Jean Tinguely (1925–1991) created kinetic sculptures usually made from industrial junk. They were hallucinatory and fabulous machines which performed unpredictably until they inevitably met a tragic fate, which was often to self-destruct. He constructed his 'Homage to New York' in the sculpture garden of the Museum of Modern Art in New York in 1960. This 23-foot-high (7.0 m) and 27-foot-long (8.2 m) mechanism performed, then self-destructed as planned in an epic and heroic manner.

Due in part to the many variables and complications associated with the production of performances of this kind, they have historically been just as likely to be "underground" affairs as officially sanctioned events. San Francisco's Survival Research Laboratories is considered to be the pioneer of the 'spectacle' form of underground robotic art.

David Karave's robotics and fire artwork, Home Automation , is an animatronic theatre performance, with themes of propaganda and peace. This robotic artwork was created over 3 years, by more than 30 artists in the USA and Canada. The project has toured across the United States, and was shown at the Tennessee Bonnaroo festival with A.S.S. The Art of Such N Such, to a crowd of approximately 80,000 giving the show perhaps the largest singular audience in the history of robotic art. In 'Home Automation' a family of lifesize aluminum animatronic crash test dummies musically self destruct, as they watch color code threat alerts on their projected home TV. The robot family's heads finally ignite into circuit breaking flames.

Two San Francisco-based performance ensembles, Frank Garvey's Omnicircus and Chico MacMurtrie's Amorphic Robot Works, were among the first expressions of integrated robotic music-theatrical performance, with human actors, dancers and musicians joining the mechanical performers (Amorphic later moved to NYC).

The Robotic Ensemble of the OmniCircus is a robot red-light district, a life-sized troupe of mechanical beggars, hookers, junkies and street-preachers who appear in OmniCircus stage shows and movies and engage in cyborg guerilla theater on the city streets. The San Francisco Bay Area has been the home and/or origin of many other mechanical performance ensembles and artists, including Matt Heckert's Mechanical Sound Orchestra, Kal Spelletich's Seemen, Carl Pisaturo, and Alan Rath, making the SF Bay Area a nexus of robotic art.

Robotic art exhibitions

- Since 2002 ArtBots has put on robotic art exhibitions featuring the work of robotics artists from around the world. Participants in each show are selected from responses to an open call for works; works are selected to represent a broad and inclusive cross-section of the tremendous range of creative art and robotics activity.

Chapter- 2

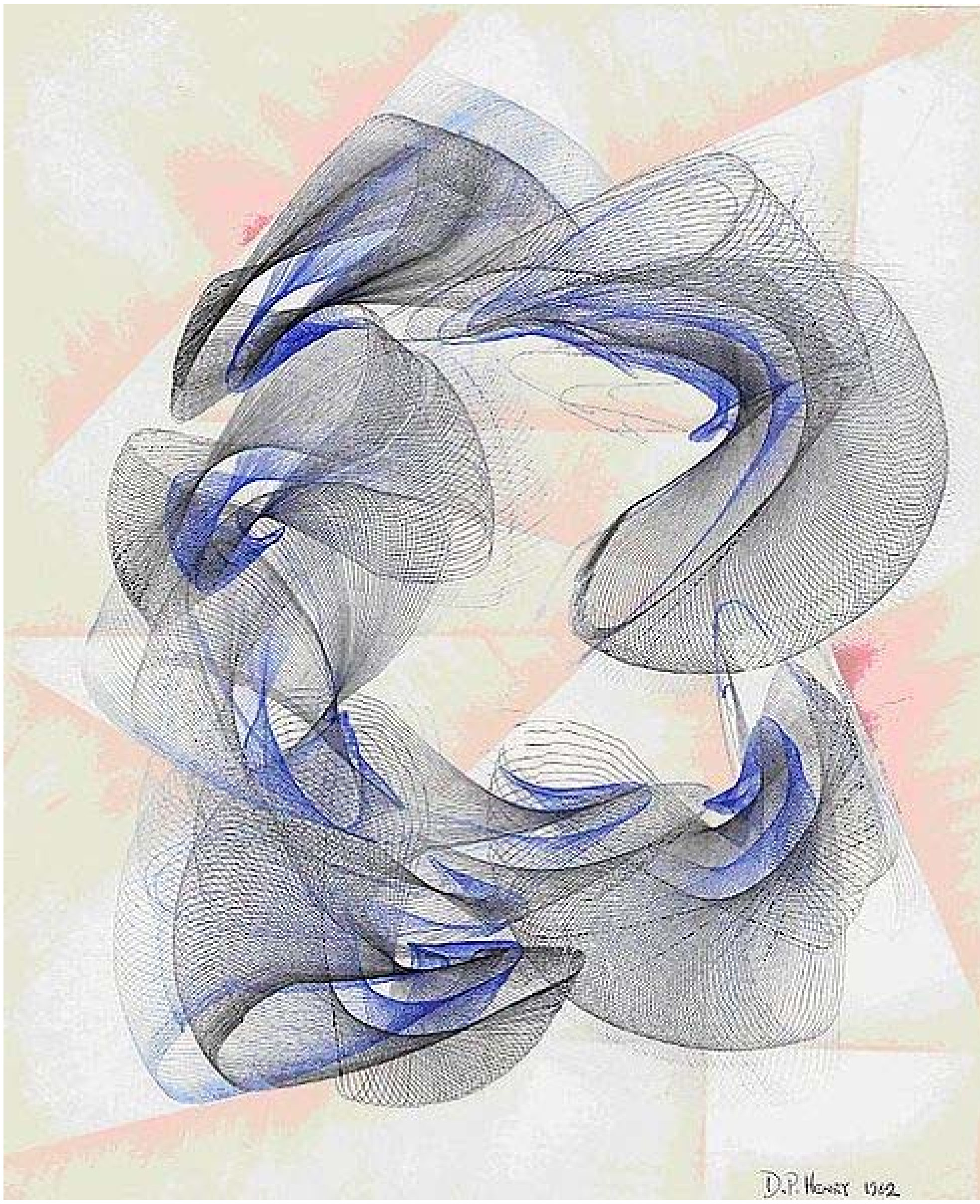
Technological Arts & their Application in Robotics

Computer art

Arambilet: *Dots on the I's*, D-ART 2009 Online Digital Art Gallery, exhibited at IV09 and CG09 computer Graphics conferences, at Pompeu Fabra University, Barcelona; Tianjin University, China; Permanent Exhibition at the London South Bank University

Computer art is any art in which computers play a role in production or display of the artwork. Such art can be an image, sound, animation, video, CD-ROM, DVD-ROM, videogame, web site, algorithm, performance or gallery installation. Many traditional disciplines are now integrating digital technologies and, as a result, the lines between traditional works of art and new media works created using computers has been blurred. For instance, an artist may combine traditional painting with algorithm art and other digital techniques. As a result, defining computer art by its end product can thus be difficult. Computer art is by its nature evolutionary since changes in technology and software directly affect what is possible. Notable artists in this vein include James Faure Walker, Manfred Mohr, Ronald Davis, Joseph Nechvatal, Matthias Groebel, George Grie, Olga Kisseleva, John Lansdown and Perry Welman.

History



Picture by drawing machine 1, Desmond Paul Henry, c.1960s

By the mid-1960s, most individuals involved in the creation of computer art were in fact engineers and scientists because they had access to the only computing resources available at university scientific research labs. Many artists tentatively began to explore the emerging computing technology for use as a creative tool. In the summer of 1962, Dr. A. Michael Noll programmed a digital computer at Bell Telephone Laboratories in Murray Hill, New Jersey to generate visual patterns solely for artistic purposes . His later computer-generated patterns simulated paintings by Piet Mondrian and Bridget Riley and become classics. Noll also used the patterns to investigate aesthetic preferences in the mid 1960s.

Computer art dates back to at least 1960, with the invention of the Henry Drawing Machine by Desmond Paul Henry. His work was shown at the Reid Gallery in London in 1962, after his machine-generated art won him the privilege of a one-man exhibition. In 1963 Joan Shogren of San Jose State University wrote a computer program based on artistic principles, resulting in an early public showing of computer art in San Jose, California on May 6, 1963.

The first two exhibitions of computer art were held in 1965- Computer-Generated Pictures, April 1965, at the Howard Wise Gallery in New York, and Generative Computergrafik, February 1965, at the Technische Hochschule in Stuttgart, Germany. The Stuttgart exhibit featured work by Georg Nees; the New York exhibit featured work by Bela Julesz and A. Michael Noll. Note the names of these expositions, not mentioning the word 'art', because these 'generated pictures' were not yet seen as such. A third exhibition was put up in November 1965 at Galerie Wendelin Niedlich in Stuttgart, Germany, showing works by Frieder Nake and Georg Nees.

In 1968, the Institute of Contemporary Arts (ICA) in London hosted one of the most influential early exhibitions of computer art- Cybernetic Serendipity. The exhibition included many of whom often regarded as the first true digital artists, Nam June Paik, Frieder Nake, Leslie Mezei, Georg Nees, A. Michael Noll, John Whitney, and Charles Csuri. One year later, the Computer Arts Society was founded, also in London.

At the time of the opening of Cybernetic Serendipity, in August 1968, a symposium was held in Zagreb, Yugoslavia, under the title "Computers and visual research". It took up the European artists movement of New Tendencies that had led to three exhibitions (in 1961, 63, and 65) in Zagreb of concrete, kinetic, and constructive art as well as op art and conceptual art. New Tendencies changed its name to "Tendencies" and continued with more symposia, exhibitions, a competition, and an international journal (bit international) until 1973.

Katherine Nash and Richard Williams published *Computer Program for Artists: ART 1* in 1970.

Xerox Corporation's Palo Alto Research Center (PARC) designed the first Graphical User Interface (GUI) in the 1970s. The first Macintosh computer is released in 1984, since then the GUI became popular. Many graphic designers quickly accepted its capacity as a creative tool.

Output devices

Formerly, technology restricted output and print results: early machines used pen-and-ink plotters to produce basic hard copy. In the 1970s, the dot matrix printer (which was much like a typewriter) was used to reproduce varied fonts and arbitrary graphics. The first animations were created by plotting all still frames sequentially on a stack of paper, with motion transfer to 16-mm film for projection. During the 1970s and 1980s, dot matrix printers were used to produce most visual output while microfilm plotters were used for most early animation.

In 1976, the inkjet printer was invented with the increase in use of personal computers. The inkjet printer is now the cheapest and most versatile option for everyday digital color output. RasterImage Processing (RIP) is typically built into the printer or supplied as a software package for the computer; it is required to achieve the highest quality output. Basic inkjet devices do not feature RIP. Instead, they rely on graphic software to rasterize images. The laser printer, though more expensive than the inkjet, is another affordable output device available today.

Graphic software

Adobe Systems, founded in 1982, developed the PostScript language and digital fonts, making drawing painting and image manipulation software popular. Adobe Illustrator, a vector drawing program based on the Bézier curve introduced in 1987 and Adobe Photoshop, written by brothers Thomas and John Knoll in 1990 were developed for use on MacIntosh computers. and compiled for DOS/Windows platforms by 1993.

Digital art

Digital art is a general term for a range of artistic works and practices that use digital technology as an essential part of the creative and/or presentation process. Since the 1970s, various names have been used to describe the process including computer art and multimedia art, and digital art is itself placed under the larger umbrella term new media art.

The impact of digital technology has transformed traditional activities such as painting, drawing and sculpture, while new forms, such as net art, digital installation art, and virtual reality, have become recognized artistic practices. More generally the term digital artist is used to describe an artist who makes use of digital technologies in the production of art. In an expanded sense, "digital art" is a term applied to contemporary art that uses the methods of mass production or digital media.

Digital production techniques in visual media

The techniques of digital art are used extensively by the mainstream media in advertisements, and by film-makers to produce special effects. Desktop publishing has had a huge impact on the publishing world, although that is more related to graphic design. It is possible that general acceptance of the value of digital art will progress in much the same way as the increased acceptance of electronically produced music over the last three decades.

Digital art can be purely computer-generated (such as fractals and algorithmic art) or taken from other sources, such as a scanned photograph or an image drawn using vector graphics software using a mouse or graphics tablet. Though technically the term may be applied to art done using other media or processes and merely scanned in, it is usually reserved for art that has been non-trivially modified by a computing process (such as a computer program, microcontroller or any electronic system capable of interpreting an input to create an output); digitized text data and raw audio and video recordings are not usually considered digital art in themselves, but can be part of the larger project of computer art and information art. Artworks are considered digital painting when created in similar fashion to non-digital paintings but using software on a computer platform and digitally outputting the resulting image as painted on canvas.

Andy Warhol created digital art with the help of Amiga, Inc. in July 1985 when he publicly introduced at Lincoln Center Amiga paint software.

Digital photography and image processing

Digital photography and digital printing is now an acceptable medium of creation and presentation by major museums and galleries. But the work of artists who produce digital paintings and digital printmakers is beginning to find acceptance, as the output capabilities advance and quality increases. Internationally, many museums are now beginning to collect digital art such as the San Jose Museum of Art and the Victoria and Albert Museum print department also has a reasonable but small collection of digital art. One reason why the established art community finds it difficult to accept digital art is the erroneous perception of digital prints being endlessly reproducible. Many artists though are erasing the relevant image file after the first print, thus making it a unique artwork.

The availability and popularity of photograph manipulation software has spawned a vast and creative library of highly modified images, many bearing little or no hint of the original image. Using electronic versions of brushes, filters and enlargers, these "neographers" produce images unattainable through conventional photographic tools. In addition, digital artists may manipulate scanned drawings, paintings, collages or lithographs, as well as using any of the above-mentioned techniques in combination. Artists also use many other sources of electronic information and programs to create their work.

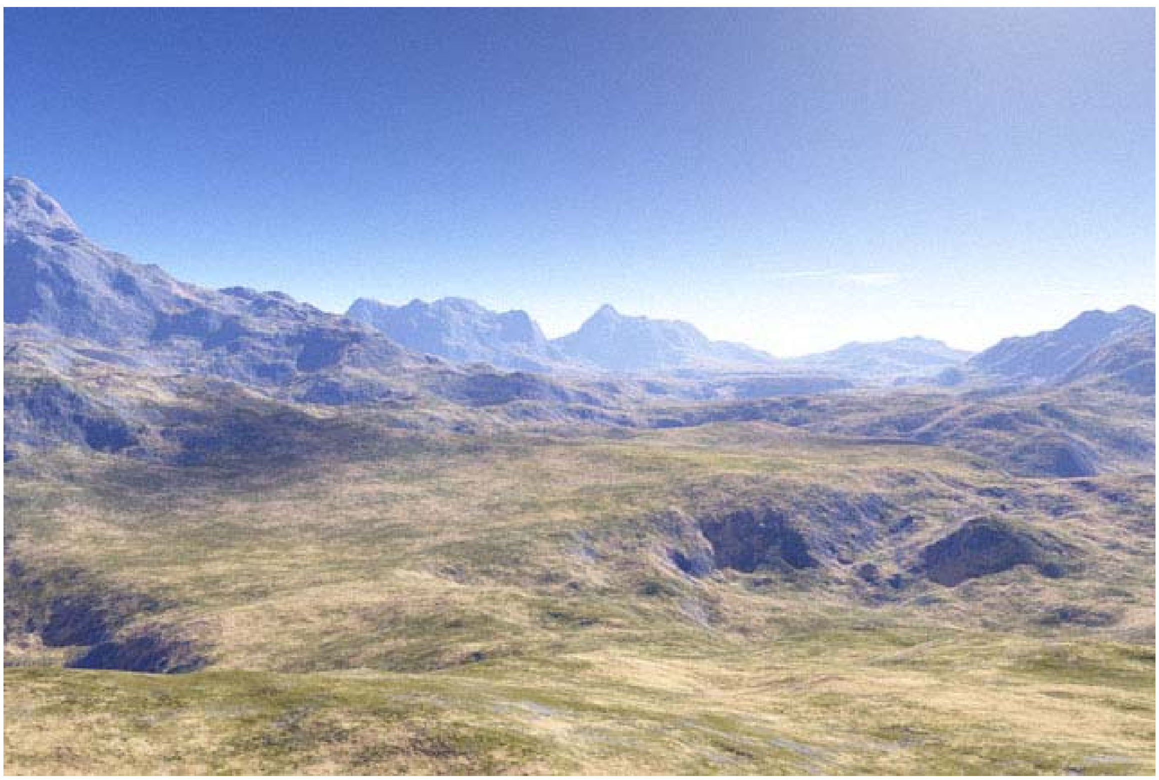
Computer-generated visual media

There are two main paradigms in computer generated imagery. The simplest is 2D computer graphics which reflect how you might draw using a pencil and a piece of paper. In this case, however, the image is on the computer screen and the instrument you draw with might be a tablet stylus or a mouse. What is generated on your screen might appear to be drawn with a pencil, pen or paintbrush. The second kind is 3D computer graphics, where the screen becomes a window into a virtual environment, where you arrange objects to be "photographed" by the computer. Typically a 2D computer graphics use raster graphics as their primary means of source data representations, whereas 3D computer graphics use vector graphics in the creation of immersive virtual reality installations. A possible third paradigm is to generate art in 2D or 3D entirely through the execution of algorithms coded into computer programs and could be considered the native art form of the computer. That is, it cannot be produced without the computer. Fractal art, Datamoshing, algorithmic art and Dynamic Painting are examples.

Computer generated 3D still imagery

3D graphics are created via the process of designing complex imagery from geometric shapes, polygons or NURBS curves to create three-dimensional shapes, objects and scenes for use in various media such as film, television, print, rapid prototyping and the special visual effects. There are many software programs for doing this. The technology can enable collaboration, lending itself to sharing and augmenting by a creative effort similar to the open source movement, and the creative commons in which users can collaborate in a project to create unique pieces of art.

Computer generated animated imagery

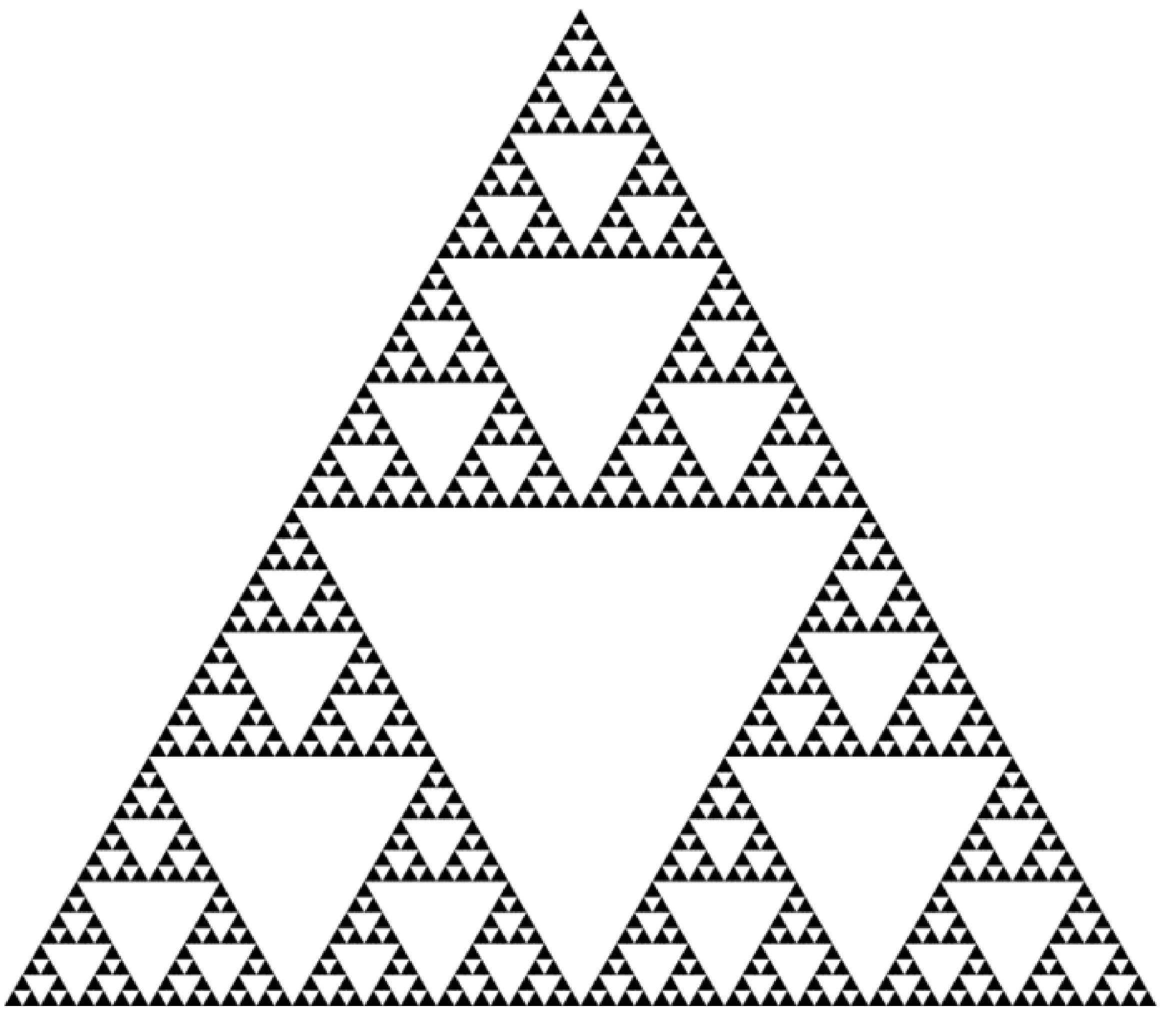


An example of a computer-generated, natural looking, static fractal landscape.

Computer-generated imagery (**CGI**) is the application of the field of computer graphics or, more specifically, 3D computer graphics to special effects in art, films, television programs, commercials, simulators and simulation generally, and printed media. The visual scenes may be either dynamic or static. The term *computer animation* refers to dynamic images.

3D computer graphics software is used to make computer-generated imagery for movies, etc. Recent availability of CGI software and increased computer speeds have allowed individual artists and small companies to produce professional grade films, games, and fine art from their home computers. This has brought about an Internet subculture with its own set of global celebrities, clichés, and technical vocabulary.

Static images and landscapes



The subdivisions of a Sierpinski triangle used for the generation of fractal backgrounds.

Not only do animated images form part of computer-generated imagery, natural looking landscapes, such as fractal landscapess are also generated via computer algorithms. A simple way to generate fractal surfaces is to use an extension of the triangular mesh method, relying on the construction of some special case a de Rham curve, e.g. midpoint displacement. For instance, the algorithm may start with a large triangle, then recursively zoom in by dividing it into 4 smaller Sierpinski triangles, then interpolate the height of each point from its nearest neighbors. The creation of a Brownian surface may be achieved not only by adding noise as new nodes are created, but by adding additional noise at multiple levels of the mesh. Thus a topographical map with varying levels of height can be created using relatively straightforward fractal algorithms. Some typical, and easy to program fractals used in CGI are the *plasma fractal* and the more dramatic *fault fractal*.

Imagery in film

CGI is used to produce special effects, figures, backgrounds, or other materials in film. For example, in *The Lord of the Rings*, director Peter Jackson uses CGI to add artificial lighting, produce massive battle scenes, and build false environments. In this film, by using CGI the character Galadriel is shown to glow bright white despite being surrounded by the deep blue tones of the setting in the scene where she shows Frodo her mirror. Then, to stage vast crowds of soldiers out of a few actors in costumes, the program Massive (for "Multiple Agent Simulation System in Virtual Environment") is used. The mountains and the trees in the background of Shire are also generated by CGI.

Interactive simulation

Simulators, particularly flight simulators, and simulation generally, make extensive use of CGI techniques for representing the outside world.

Architectural scenes

Architects use services from computer graphic firms to create a 3-dimensional models for both the customers and builders. It can be more accurate than traditional drawings. Architectural animation can also be used to see the possible relationship the building will have in relation to the environment and its surrounding buildings.

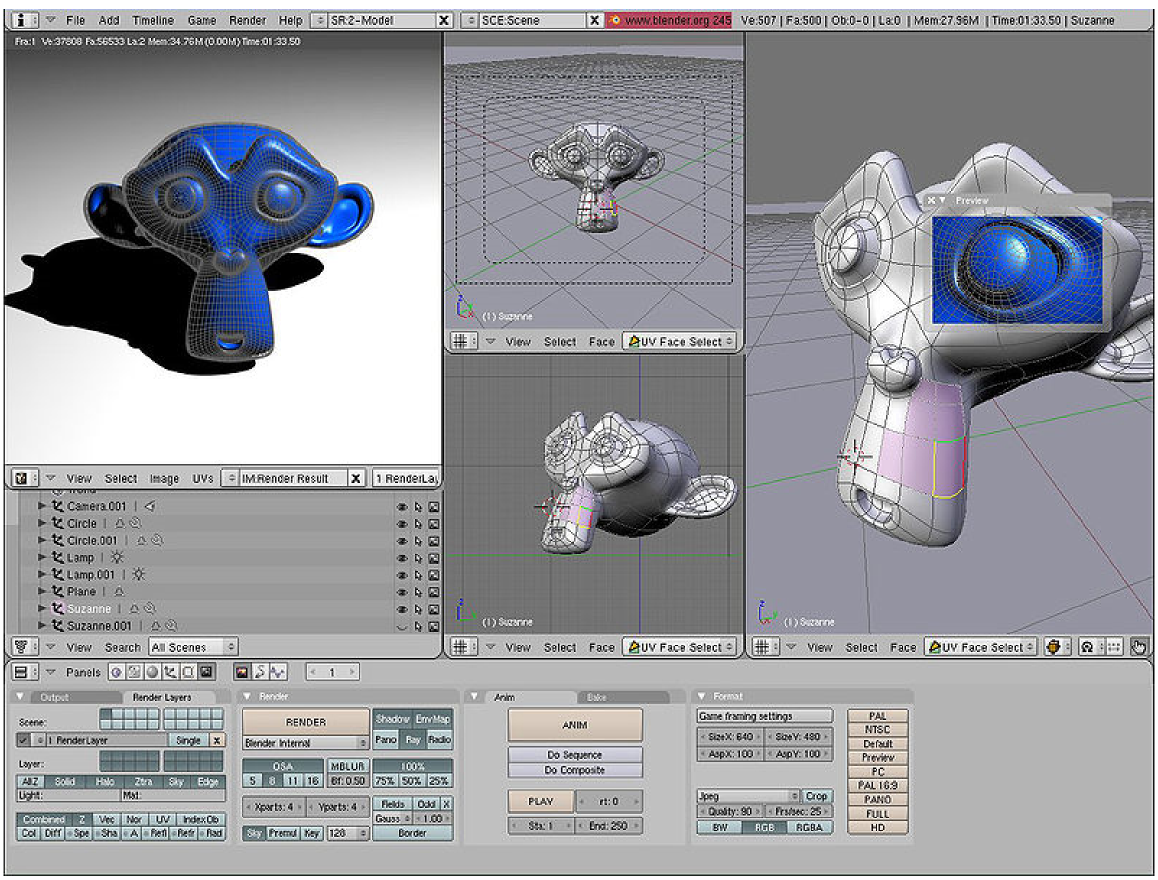
Digital installation art

Digital installation art constitutes a broad field of activity and incorporates many forms. Some resemble video installations, particularly large scale works involving projections and live video capture. By using projection techniques that enhance an audiences impression of sensory envelopment, many digital installations attempt to create immersive environments. Others go even further and attempt to facilitate a complete immersion in virtual realms. This type of installation is generally site specific, scalable, and without fixed dimensionality, meaning it can be reconfigured to accommodate different presentation spaces.

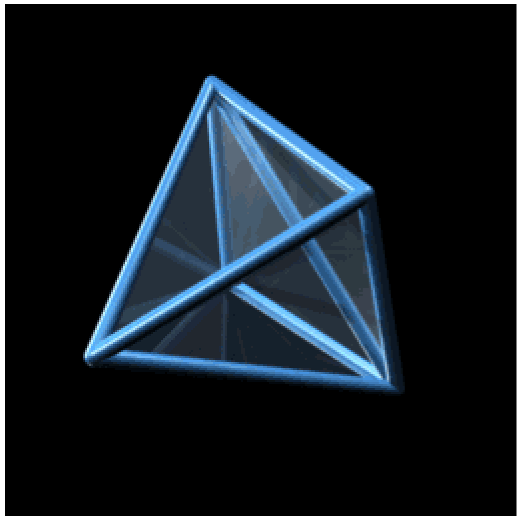
Noah Wardrip-Fruin's interactive new media art piece entitled "Screen is an example of digital installation art. To view and interact with the piece, a user first enters a room, called the "Cave," which is a virtual reality display area with four walls surrounding the participant. White memory texts appear on the background of black walls. Through bodily interaction, such as using one's hand, a user can move and bounce the text around the walls. The words can be made into sentences and eventually begin to "peel" off and move more rapidly around the user, creating a heightening sense of misplacement.

"In addition to creating a new form of bodily interaction with text through its play, Screen moves the player through three reading experiences — beginning with the familiar, stable, page-like text on the walls, followed by the word-by-word reading of peeling and hitting (where attention is focused), and with more peripheral awareness of the arrangements of flocking words and the new (often neologistic) text being assembled on the walls. Screen was first shown in 2003 as part of the Boston Cyberarts Festival (in the Cave at Brown University) and documentation of it has since been featured at The Iowa Review Web, presented at SIGGRAPH 2003, included in Alt+Ctrl: a festival of independent and alternative games, published in the DVD magazines Aspect and Chaise, as well as in readings in the Hammer Museum's HyperText series, at ACM Hypertext 2004, and in other venues."

Computer graphics



A Blender 2.45 screenshot.



A 2D projection of a 3D projection of a 4D Pentachoron performing a double rotation about two orthogonal planes.

Computer graphics are graphics created using computers and, more generally, the representation and manipulation of image data by a computer.

The development of computer graphics, has made computers easier to interact with, and better for understanding and interpreting many types of data. Developments in computer graphics have had a profound impact on many types of media and have revolutionized animation, movies and the video game industry.

Overview

The term computer graphics has been used in a broad sense to describe "almost everything on computers that is not text or sound". Typically, the term *computer graphics* refers to several different things:

- the representation and manipulation of image data by a computer
- the various technologies used to create and manipulate images
- the images so produced, and
- the sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content, see study of computer graphics

Today, computers and computer-generated images touch many aspects of daily life. Computer imagery is found on television, in newspapers, for example in weather reports, or for example in all kinds of medical investigation and surgical procedures. A well-constructed graph can present complex statistics in a form that is easier to understand and interpret. In the media "such graphs are used to illustrate papers, reports, theses", and other presentation material.

Many powerful tools have been developed to visualize data. Computer generated imagery can be categorized into several different types: 2D, 3D, 5D, and animated graphics. As technology has improved, 3D computer graphics have become more common, but 2D computer graphics are still widely used. Computer graphics has emerged as a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Over the past decade, other specialized fields have been developed like information visualization, and scientific visualization more concerned with "the visualization of three dimensional phenomena (architectural, meteorological, medical, biological, etc.), where the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component".

Initial Development of Computer Graphics

The advance in computer graphics was to come from one MIT student, Ivan Sutherland. In 1961 Sutherland created another computer drawing program called Sketchpad. Using a light pen, Sketchpad allowed one to draw simple shapes on the computer screen, save them and even recall them later. The light pen itself had a small photoelectric cell in its tip. This cell emitted an electronic pulse whenever it was placed in front of a computer screen and the screen's electron gun fired directly at it. By simply timing the electronic pulse with the current location of the electron gun, it was easy to pinpoint exactly where the pen was on the screen at any given moment. Once that was determined, the computer could then draw a cursor at that location.

Sutherland seemed to find the perfect solution for many of the graphics problems he faced. Even today, many standards of computer graphics interfaces got their start with this early Sketchpad program. One example of this is in drawing constraints. If one wants to draw a square for example, s/he doesn't have to worry about drawing four lines perfectly to form the edges of the box. One can simply specify that s/he wants to draw a box, and then specify the location and size of the box. The software will then construct a perfect box, with the right dimensions and at the right location. Another example is that Sutherland's software modeled objects - not just a picture of objects. In other words, with a model of a car, one could change the size of the tires without affecting the rest of the car. It could stretch the body of the car without deforming the tires.

These early computer graphics were Vector graphics, composed of thin lines whereas modern day graphics are Raster based using pixels. The difference between vector graphics and raster graphics can be illustrated with a shipwrecked sailor. He creates an SOS sign in the sand by arranging rocks in the shape of the letters "SOS." He also has some brightly colored rope, with which he makes a second "SOS" sign by arranging the rope in the shapes of the letters. The rock SOS sign is similar to raster graphics. Every pixel has to be individually accounted for. The rope SOS sign is equivalent to vector graphics. The computer simply sets the starting point and ending point for the line and perhaps bend it a little between the two end points. The disadvantages to vector files are that they cannot represent continuous tone images and they are limited in the number of colors available. Raster formats on the other hand work well for continuous tone images and can reproduce as many colors as needed.

Also in 1961 another student at MIT, Steve Russell, created the first video game, Spacewar. Written for the DEC PDP-1, Spacewar was an instant success and copies started flowing to other PDP-1 owners and eventually even DEC got a copy. The engineers at DEC used it as a diagnostic program on every new PDP-1 before shipping it. The sales force picked up on this quickly enough and when installing new units, would run the world's first video game for their new customers.

E. E. Zajac, a scientist at Bell Telephone Laboratory (BTL), created a film called "Simulation of a two-giro gravity attitude control system" in 1963. In this computer generated film, Zajac showed how the attitude of a satellite could be altered as it orbits the Earth. He created the animation on an IBM 7090 mainframe computer. Also at BTL, Ken Knowlton, Frank Sindon and Michael Noll started working in the computer graphics field. Sindon created a film called Force, Mass and Motion illustrating Newton's laws of motion in operation. Around the same time, other scientists were creating computer graphics to illustrate their research. At Lawrence Radiation Laboratory, Nelson Max created the films, "Flow of a Viscous Fluid" and "Propagation of Shock Waves in a Solid Form." Boeing Aircraft created a film called "Vibration of an Aircraft."

It wasn't long before major corporations started taking an interest in computer graphics. TRW, Lockheed-Georgia, General Electric and Sperry Rand are among the many companies that were getting started in computer graphics by the mid 1960's. IBM was quick to respond to this interest by releasing the IBM 2250 graphics terminal, the first commercially available graphics computer.

Ralph Baer, a supervising engineer at Sanders Associates, came up with a home video game in 1966 that was later licensed to Magnavox and called the Odyssey. While very simplistic, and requiring fairly inexpensive electronic parts, it allowed the player to move points of light around on a screen. It was the first consumer computer graphics product.

Also in 1966, Sutherland at MIT invented the first computer controlled head-mounted display (HMD). Called the Sword of Damocles because of the hardware required for support, it displayed two separate wireframe images, one for each eye. This allowed the viewer to see the computer scene in stereoscopic 3D. After receiving his Ph.D. from MIT, Sutherland became Director of Information Processing at ARPA (Advanced Research Projects Agency), and later became a professor at Harvard.

Dave Evans was director of engineering at Bendix Corporation's computer division from 1953 to 1962, after which he worked for the next five years as a visiting professor at Berkeley. There he continued his interest in computers and how they interfaced with people. In 1968 the University of Utah recruited Evans to form a computer science program, and computer graphics quickly became his primary interest. This new department would become the world's primary research center for computer graphics.

In 1967 Sutherland was recruited by Evans to join the computer science program at the University of Utah. There he perfected his HMD. Twenty years later, NASA would re-discover his techniques in their virtual reality research. At Utah, Sutherland and Evans were highly sought after consultants by large companies but they were frustrated at the lack of graphics hardware available at the time so they started formulating a plan to start their own company.

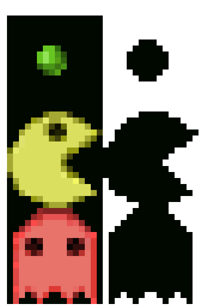
A student by the name of Edwin Catmull started at the University of Utah in 1970 and signed up for Sutherland's computer graphics class. Catmull had just come from The Boeing Company and had been working on his degree in physics. Growing up on Disney, Catmull loved animation yet quickly discovered that he didn't have the talent for drawing. Now Catmull (along with many others) saw computers as the natural progression of animation and they wanted to be part of the revolution. The first animation that Catmull saw was his own. He created an animation of his hand opening and closing. It became one of his goals to produce a feature length motion picture using computer graphics. In the same class, Fred Parke created an animation of his wife's face. Because of Evan's and Sutherland's presence, UU was gaining quite a reputation as the place to be for computer graphics research so Catmull went there to learn 3D animation.

As the UU computer graphics laboratory was attracting people from all over, John Warnock was one of those early pioneers; he would later found Adobe Systems and create a revolution in the publishing world with his PostScript page description language. Tom Stockham led the image processing group at UU which worked closely with the computer graphics lab. Jim Clark was also there; he would later found Silicon Graphics, Inc.

The first major advance in 3D computer graphics was created at UU by these early pioneers, the hidden-surface algorithm. In order to draw a representation of a 3D object on the screen, the computer must determine which surfaces are "behind" the object from the viewer's perspective, and thus should be "hidden" when the computer creates (or renders) the image.

Image types

2D computer graphics



Raster graphic sprites (left) and masks (right)

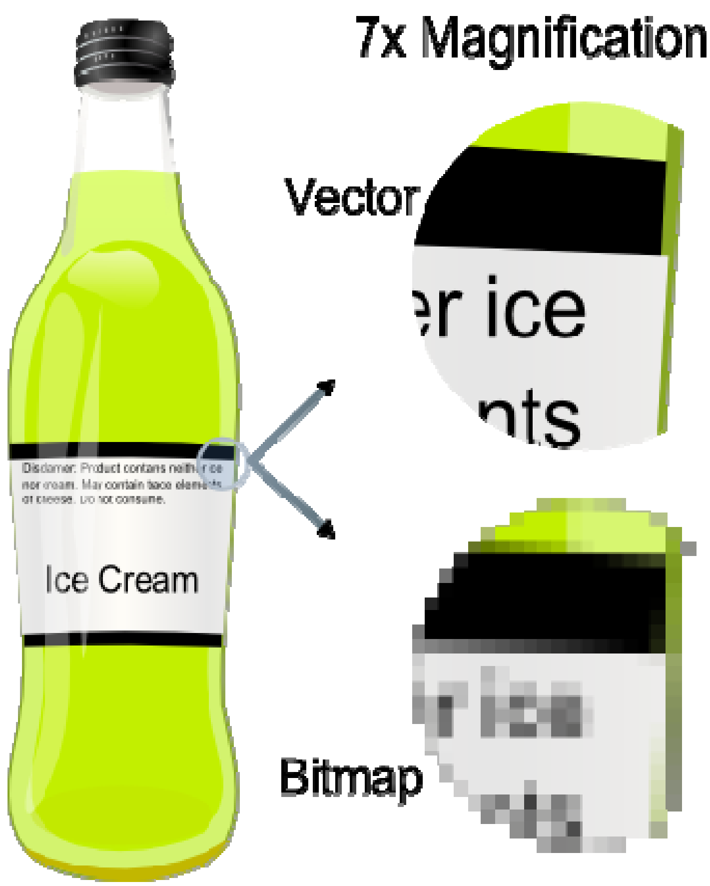
2D computer graphics are the computer-based generation of digital images—mostly from two-dimensional models, such as 2D geometric models, text, and digital images, and by techniques specific to them. The word may stand for the branch of computer science that comprises such techniques, or for the models themselves.

2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawing, advertising, etc.. In those applications, the two-dimensional image is not just a representation of a real-world object, but an independent artifact with added semantic value; two-dimensional models are therefore preferred, because they give more direct control of the image than 3D computer graphics, whose approach is more akin to photography than to typography.

Pixel art

Pixel art is a form of digital art, created through the use of raster graphics software, where images are edited on the pixel level. Graphics in most old (or relatively limited) computer and video games, graphing calculator games, and many mobile phone games are mostly pixel art.

Vector graphics



Example showing effect of vector graphics versus raster (bitmap) graphics.

Vector graphics formats are complementary to raster graphics, which is the representation of images as an array of pixels, as it is typically used for the representation of photographic images. There are instances when working with vector tools and formats is best practice, and instances when working with raster tools and formats is best practice. There are times when both formats come together. An understanding of the advantages and limitations of each technology and the relationship between them is most likely to result in efficient and effective use of tools.

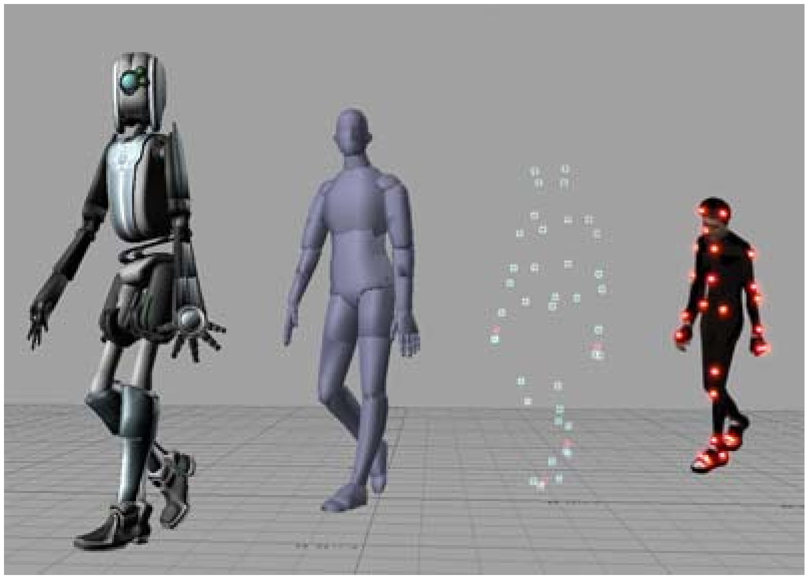
3D computer graphics

3D computer graphics in contrast to 2D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. Such images may be for later display or for real-time viewing.

Despite these differences, 3D computer graphics rely on many of the same algorithms as 2D computer vector graphics in the wire frame model and 2D computer raster graphics in the final rendered display. In computer graphics software, the distinction between 2D and 3D is occasionally blurred; 2D applications may use 3D techniques to achieve effects such as lighting, and primarily 3D may use 2D rendering techniques.

3D computer graphics are often referred to as 3D models. Apart from the rendered graphic, the model is contained within the graphical data file. However, there are differences. A 3D model is the mathematical representation of any three-dimensional object. A model is not technically a graphic until it is visually displayed. Due to 3D printing, 3D models are not confined to virtual space. A model can be displayed visually as a two-dimensional image through a process called *3D rendering,* or used in non-graphical computer simulations and calculations. There are some 3D computer graphics software for users to create 3D images.

Computer animation



An example of Computer animation produced using Motion capture

Computer animation is the art of creating moving images via the use of computers. It is a subfield of computer graphics and animation. Increasingly it is created by means of 3D computer graphics, though 2D computer graphics are still widely used for stylistic, low bandwidth, and faster real-time rendering needs. Sometimes the target of the animation is the computer itself, but sometimes the target is another medium, such as film. It is also referred to as CGI (Computer-generated imagery or computer-generated imaging), especially when used in films.

Virtual entities may contain and be controlled by assorted attributes, such as transform values (location, orientation, and scale) stored in an object's transformation matrix. Animation is the change of an attribute over time. Multiple methods of achieving animation exist; the rudimentary form is based on the creation and editing of keyframes, each storing a value at a given time, per attribute to be animated. The 2D/3D graphics software will interpolate between keyframes, creating an editable curve of a value mapped over time, resulting in animation. Other methods of animation include procedural and expression-based techniques: the former consolidates related elements of animated entities into sets of attributes, useful for creating particle effects and crowd simulations; the latter allows an evaluated result returned from a user-defined logical expression, coupled with mathematics, to automate animation in a predictable way (convenient for controlling bone behavior beyond what a hierarchy offers in skeletal system set up).

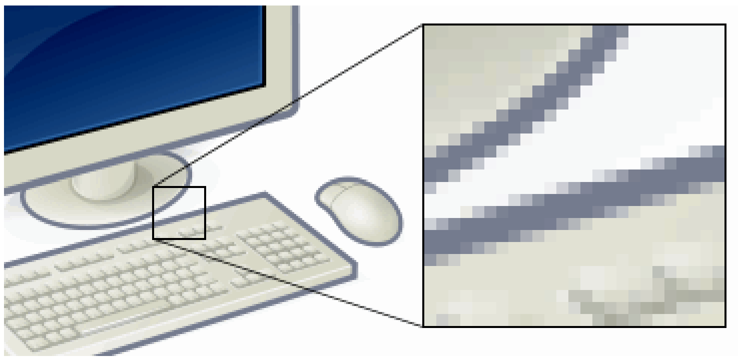
To create the illusion of movement, an image is displayed on the computer screen then quickly replaced by a new image that is similar to the previous image, but shifted slightly. This technique is identical to the illusion of movement in television and motion pictures.

Concepts and principles

Images are typically produced by optical devices;such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

A digital image is a representation of a two-dimensional image in binary format as a sequence of ones and zeros. Digital images include both vector images and raster images, but raster images are more commonly used.

Pixel



In the enlarged portion of the image individual pixels are rendered as squares and can be easily seen.

In digital imaging, a pixel (or picture element) is a single point in a raster image. Pixels are normally arranged in a regular 2-dimensional grid, and are often represented using dots or squares. Each pixel is a sample of an original image, where more samples typically provide a more accurate representation of the original. The intensity of each pixel is variable; in color systems, each pixel has typically three components such as red, green, and blue.

Graphics

Graphics are visual presentations on some surface, such as a wall, canvas, computer screen, paper, or stone to brand, inform, illustrate, or entertain. Examples are photographs, drawings, line art, graphs, diagrams, typography, numbers, symbols, geometric designs, maps, engineering drawings, or other images. Graphics often combine text, illustration, and color. Graphic design may consist of the deliberate selection, creation, or arrangement of typography alone, as in a brochure, flier, poster, web site, or book without any other element. Clarity or effective communication may be the objective, association with other cultural elements may be sought, or merely, the creation of a distinctive style.

Arambilet: *Dots on the I's*, D-ART 2009 Online Digital Art Gallery, exhibited at IV09 and CG09 computer Graphics conferences, at Pompeu Fabra University, Barcelona; Tianjin University, China; Permanent Exhibition at the London South Bank University

Rendering

Rendering is the process of generating an image from a model (or models in what collectively could be called a *scene* file), by means of computer programs. A scene file contains objects in a strictly defined language or data structure; it would contain geometry, viewpoint, texture, lighting, and shading information as a description of the virtual scene. The data contained in the scene file is then passed to a rendering program to be processed and output to a digital image or raster graphics image file. The rendering program is usually built into the computer graphics software, though others are available as plug-ins or entirely separate programs. The term "rendering" may be by analogy with an "artist's rendering" of a scene. Though the technical details of rendering methods vary, the general challenges to overcome in producing a 2D image from a 3D representation stored in a scene file are outlined as the graphics pipeline along a rendering device, such as a GPU. A GPU is a purpose-built device able to assist a CPU in performing complex rendering calculations. If a scene is to look relatively realistic and predictable under virtual lighting, the rendering software should solve the rendering equation. The rendering equation doesn't account for all lighting phenomena, but is a general lighting model for computer-generated imagery. 'Rendering' is also used to describe the process of calculating effects in a video editing file to produce final video output.

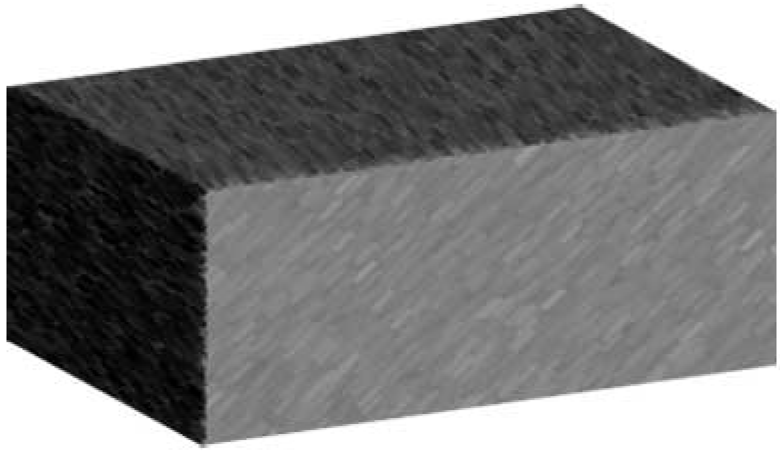
3D projection

3D projection is a method of mapping three dimensional points to a two dimensional plane. As most current methods for displaying graphical data are based on planar two dimensional media, the use of this type of projection is widespread, especially in computer graphics, engineering and drafting.

Ray tracing

Ray tracing is a technique for generating an image by tracing the path of light through pixels in an image plane. The technique is capable of producing a very high degree of photorealism; usually higher than that of typical scanline rendering methods, but at a greater computational cost.

Shading



Example of shading.

Shading refers to depicting depth in 3D models or illustrations by varying levels of darkness. It is a process used in drawing for depicting levels of darkness on paper by applying media more densely or with a darker shade for darker areas, and less densely or with a lighter shade for lighter areas. There are various techniques of shading including cross hatching where perpendicular lines of varying closeness are drawn in a grid pattern to shade an area. The closer the lines are together, the darker the area appears. Likewise, the farther apart the lines are, the lighter the area appears. The term has been recently generalized to mean that shaders are applied.

Texture mapping

Texture mapping is a method for adding detail, surface texture, or colour to a computer-generated graphic or 3D model. Its application to 3D graphics was pioneered by Dr Edwin Catmull in 1974. A texture map is applied (mapped) to the surface of a shape, or polygon. This process is akin to applying patterned paper to a plain white box. Multitexturing is the use of more than one texture at a time on a polygon. Procedural textures (created from adjusting parameters of an underlying algorithm that produces an output texture), and bitmap textures (created in an image editing application) are, generally speaking, common methods of implementing texture definition from a 3D animation program, while intended placement of textures onto a model's surface often requires a technique known as UV mapping.

Anti-aliasing

Rendering resolution-independent entities (such as 3D models) for viewing on a raster (pixel-based) device such as a LCD display or CRT television inevitably causes aliasing artifacts mostly along geometric edges and the boundaries of texture details; these artifacts are informally called "jaggies". Anti-aliasing methods rectify such problems, resulting in imagery more pleasing to the viewer, but can be somewhat computationally expensive. Various anti-aliasing algorithms (such as supersampling) are able to be employed, then customized for the most efficient rendering performance versus quality of the resultant imagery; a graphics artist should consider this trade-off if anti-aliasing methods are to be used. A preanti-aliased bitmap texture being displayed on a screen (or screen location) at a resolution different than the resolution of the texture itself (such as a textured model in the distance from the virtual camera) will exhibit aliasing artifacts, while any procedurally-defined texture will always show aliasing artifacts as they are resolution-independent; techniques such as mipmapping and texture filtering help to solve texture-related aliasing problems.

Volume rendering

Volume rendered CT scan of a forearm with different colour schemes for muscle, fat, bone, and blood.

Volume rendering is a technique used to display a 2D projection of a 3D discretely sampled data set. A typical 3D data set is a group of 2D slice images acquired by a CT or MRI scanner.

Usually these are acquired in a regular pattern (e.g., one slice every millimeter) and usually have a regular number of image pixels in a regular pattern. This is an example of a regular volumetric grid, with each volume element, or voxel represented by a single value that is obtained by sampling the immediate area surrounding the voxel.

3D modeling

3D modeling is the process of developing a mathematical, wireframe representation of any three-dimensional object, called a "3D model", via specialized software. Models may be created automatically or manually; the manual modeling process of preparing geometric data for 3D computer graphics is similar to plastic arts such as sculpting. 3D models may be created using multiple approaches: use of NURBS curves to generate accurate and smooth surface patches, polygonal mesh modeling (manipulation of faceted geometry), or polygonal mesh subdivision (advanced tessellation of polygons, resulting in smooth surfaces similar to NURBS models). A 3D model can be displayed as a two-dimensional image through a process called *3D rendering*, used in a computer simulation of physical phenomena, or animated directly for other purposes. The model can also be physically created using 3D Printing devices.

Pioneers in graphic design

Charles Csuri

Charles Csuri is a pioneer in computer animation and digital fine art and created the first computer art in 1964. Csuri was recognized by *Smithsonian* as the father of digital art and computer animation, and as a pioneer of computer animation by the Museum of Modern Art (MoMA) and Association for Computing Machinery-SIGGRAPH.

Donald P. Greenberg

Donald P. Greenberg is a leading innovator in computer graphics. Greenberg has authored hundreds of articles and served as a teacher and mentor to many prominent computer graphic artists, animators, and researchers such as Robert L. Cook, Marc Levoy, and Wayne Lytle. Many of his former students have won Academy Awards for technical achievements and several have won the SIGGRAPH Achievement Award. Greenberg was the founding director of the NSF Center for Computer Graphics and Scientific Visualization.

A. Michael Noll

Noll was one of the first researchers to use a digital computer to create artistic patterns and to formalize the use of random processes in the creation of visual arts. He began creating digital computer art in 1962, making him one of the earliest digital computer artists. In 1965, Noll along with Frieder Nake and Georg Nees were the first to publicly exhibit their computer art. During April 1965, the Howard Wise Gallery exhibited Noll's computer art along with random-dot patterns by Bela Julesz.

- Jim Blinn
- Arambilet
- Benoît B. Mandelbrot
- Henri Gouraud
- Bui Tuong Phong
- Pierre Bézier
- Paul de Casteljau

- Daniel J. Sandin
- Alvy Ray Smith
- Ton Roosendaal
- Ivan Sutherland
- Steve Russell

The study of computer graphics

The study of computer graphics is a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Although the term often refers to three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing.

As an academic discipline, computer graphics studies the manipulation of visual and geometric information using computational techniques. It focuses on the *mathematical* and *computational* foundations of image generation and processing rather than purely aesthetic issues. Computer graphics is often differentiated from the field of visualization, although the two fields have many similarities.

Interactive art

The Tunnel under the Atlantic (1995), Maurice Benayoun, Virtual Reality Interactive Installation : a link between Paris and Montréal

Interactive art is a form of installation-based art that involves the spectator in a way that allows the art to achieve its purpose. Some installations achieve this by letting the observer walk in, on, and around them.

Works of this kind of art frequently feature computers and sensors to respond to motion, heat, meteorological changes or other types of input their makers programmed them to respond to. Most examples of virtual Internet art and electronic art are highly interactive. Sometimes, visitors are able to navigate through a hypertext environment; some works accept textual or visual input from outside; sometimes an audience can influence the course of a performance or can even participate in it.

Though some of the earliest examples of interactive art have been dated back to the 1920s, most digital art didn't make its official entry into the world of art until the late 1990s. Since this debut, countless museums and venues have been increasingly accommodating digital and interactive art into their productions. This budding genre of art is continuing to grow and evolve in a somewhat rapid manner through internet social sub-culture in one hand, and large scale urban installations in the other hand.

Interactivity in art

Interactive art is a genre of art in which the viewers participate in some way by providing an input in order to determine the outcome. Unlike traditional art forms wherein the interaction of the spectator is merely a mental event, interactivity allows for various types of navigation, assembly, and/or contribution to an artwork, which goes far beyond purely psychological activity. Interactivity as a medium produces meaning.

Interactive art installations are generally computer-based and frequently rely on sensors, which gauge things such as temperature, motion, proximity, and other meteorological phenomena that the maker has programmed in order to elicit responses based on participant action. In interactive artworks, both the audience and the machine work together in dialogue in order to produce a completely unique artwork for each audience to observe. However, not all observers visualize the same picture. Because it is interactive art, each observer makes their own interpretation of the artwork and it may be completely different than another observer's views.

Interactive art can be distinguished from Generative art in that it constitutes a dialogue between the artwork and the participant; specifically, the participant has agency, or the ability, even in an unintentional manner, to act upon the artwork and is furthermore invited to do so within the context of the piece, i.e. the work affords the interaction. More often, we can consider that the work takes its visitor into account. In an increasing number of cases an installation can be defined as a responsive environment, especially those created by architects and designers. By contrast, Generative Art, which may be interactive, but not responsive per se, tends to be a monologue - the artwork may change or evolve in the presence of the viewer, but the viewer may not be invited to engage in the reaction but merely enjoy it.

History

According to the new media artist Maurice Benayoun, the first piece of interactive art should be the work done by Parrhasius during his art contest with Zeuxis described by Pliny, in the fifth century B.C. when Zeuxis tried to unveil the painted curtain. The work takes its meaning from Zeuxis gesture and wouldn't exist without it. Zeuxis, by its gesture, became part of Parrhasius' work. This shows that the specificity of interactive art resides often less in the use of computers than in the quality of proposed "situations" and the "Other's" involvement in the process of sensemaking. Nevertheless computers and real time computing made the task easier and opened the field of virtuality- the potential emergence of unexpected (although possibly pre-written) futures- to contemporary arts.

Some of the earliest examples of interactive art were created as early as the 1920s. An example is Marcel Duchamp's piece named *Rotary Glass Plates*. The artwork required the viewer to turn on the machine and stand at a distance of one meter.

The present idea of interactive art began to flourish more in the 1960s for partly political reasons. At the time, many people found it inappropriate for artists to carry the only creative power within their works. Those artists who held this view wanted to give the audience their own part of this creative process. Aside from this "political" view, it was also current wisdom that interaction and engagement had a positive part to play within the creative process.

In the 1970s artists began to use new technology such as video and satellites to experiment with live performances and interactions through the direct broadcast of video and audio.

Interactive art became a large phenomenon due to the advent of computer based interactivity in the 1990s. Along with this came a new kind of art-experience. Audience and machine were now able to more easily work together in dialogue in order to produce a unique artwork for each audience. In the late 1990s, museums and galleries began increasingly incorporating the art form in their shows, some even dedicating entire exhibitions to it. This continues today and is only expanding due to increased communications through digital media.

A hybrid emerging discipline drawing on the combined interests of specific artists and architects has been created in the last 10–15 years. Disciplinary boundaries have blurred, and significant number of architects and interactive designers have joined electronic artists in the creation of new, custom-designed interfaces and evolutions in techniques for obtaining user input (such as dog vision, alternative sensors, voice analysis, etc.); forms and tools for information display (such as video projection, lasers, robotic and mechatronic actuators, led lighting etc.); modes for human-human and human-machine communication (through the Internet and other telecommunications networks); and to the development of social contexts for interactive systems (such as utilitarian tools, formal experiments, games and entertainment, social critique, and political liberation).

Forms

There are many different forms of interactive art. Such forms range from interactive dance, music, and even drama. New technology, primarily computer systems and computer technology, have enabled a new class of interactive art. Examples of such interactive art are installation art, interactive architecture and interactive film.

Impact

The aesthetic impact of interactive art is more profound than expected.

Supporters of more "traditional" contemporary art saw, in the use of computers, a way to balance artistic deficiencies, some other consider that the art is not anymore in the achievement of the formal shape of the work but in the design of the rules that determine the evolution of the shape according to the quality of the dialogue.

Interactive art events and places

There are number of globally significant festivals and exhibitions of interactive and media arts. Prix Ars Electronica is a major yearly competition and exhibition that gives awards to outstanding examples of (technology-driven) interactive art. Association of Computing Machinery's Special Interest Group in Graphics (SIGGRAPH), DEAF Dutch Electronic Arts Festival, Transmediale Germany, FILE - Electronic Language International Festival Brazil, and AV Festival England, are among the others.

CAiiA, Centre for Advanced Inquiry in the Interactive Arts, first established in 1994 at the University of Wales, Newport, and later in 2003 as the Planetary Collegium, was the first doctoral and post doc research center to be established specifically for research in the interactive art field.

Interactive architecture has now been installed on and as part of building facades, in foyers, museums and large scale public spaces, including airports, in a number of global cities. A number of leading museums, for example, the National Gallery, Tate, Victoria & Albert Museum and Science Museum in London (to cite the leading UK museums active in this field) were early adoptors in the field of interactive technologies, investing in educational resources, and more latterly, in the creative use of MP3 players for visitors. In 2004 the Victoria & Albert Museum commissioned curator and author Lucy Bullivant to write Responsive Environments (2006), the first such publication of its kind. Interactive designers are frequently commissioned for museum displays; a number specialize in wearable computing.

Knowbotic Research

Knowbotic Research is a German-Swiss electronic art group, established in 1991. Its members are Yvonne Wilhelm, Christian Hübler and Alexander Tuchacek. They hold a professorship for Art and Media at the University of the Arts in Zurich.

History

Yvonne Wilhelm (born 1962), Christian Huebler (born 1962), Alexander Tuchacek (born 1962) are based in Zurich Switzerland.

The Knowbotic Research group has experimented with the intersection of technology, information and knowledge, interface, immersive virtual reality and networked agency. In their work *Simulation Mosaik Data Klaenge* from 1993, they experimented with so called intelligent agent, applications which can conglomerate diaphanous information by themselves (also called knowbots) and intelligent virtual spaces (flexible information-environments distributed in electronic networks). Knowbotic Research **KR+cF** has regularly invited people from non-art fields to participate in their projects, such as scientists, philosophers and engineers, depending on the concept of each project. In partnership with the Academy of Media Arts Cologne, Knowbotic Research KR+cF has founded *Membrane*, a laboratory for media strategies, in 1995. In their more recent work, they created media-based projects that try to intervene in the (physical or digital) public domain.

Knowbotic is a coinage that combined “knowledge” with “robot”, meaning intellectual agent on the Internet. Knowbotic has developed some projects themed on an information environment and a computer interface. Since 1998, it has become more flexible, and with these main three of them, different members from various fields such as art, science, and philosophy, have joined in each program. In 1997, it worked with Japanese art group, Canon Art Lab, in Tokyo. This project aimed at revealing the function of the city by interacting between real and virtual world.

Chapter- 3

Well Known Robotic Artists

Chico MacMurtrie

Chico MacMurtrie was born in New Mexico in 1961. He has been awarded four grants from the National Endowment for the Arts for Interdisciplinary Artists. In 1990 he received the San Francisco Bay Guardian Goldie Award.

In 1992 MacMurtrie formed Amorphic Robot Works, a group of artists and engineers working together to create robotic art performances and installations. His permanent commissioned interactive sculptures include the anthropomorphic *Urge to Stand* at Yerba Buena Gardens in San Francisco, *Fetus to Man* sculpture/clock for the city of Lille France, and *Growing Raining Tree* at Contemporary Arts Center in Cincinnati OH.

ARW's Artistic Director Chico MacMurtrie, describes his vision, "The work is an ongoing endeavor to uncover the primacy of movement and sound. Each machine is inspired or influenced, both, by modern society, and what I physically experience and sense. The whole of this input informs my ideas and work."

Leonel Moura

Leonel Moura (born on December 26, 1948 in Lisbon, Portugal) is a conceptual artist whose work shifted in the late 90’s from photo based work to Artificial Intelligence and Robotic art. Since then he has produced several Painting Robots and the Robotarium, a zoo for robots, the first of its kind in the world. RAP (Robotic Action Painter), 2006, a robot that makes drawings based on emergence and stigmergy; decides when the work is ready and signs it, is displayed as a permanent installation at the American Museum of Natural History in New York.

Robots

ArtSbot (Art Swarm Robots), 2003, comprise several small autonomous robots, called Mbots, each equipped with color detection sensors, obstacle avoidance sensors, a microcontroller and actuators, for locomotion and pen manipulation. Mbots have two distinct behaviors: the random behavior that initializes the process by activating a pen, based on a small probability (usually 2/256), whenever the color sensors read white; and the positive feed-back behavior that reinforces the color detected by the sensors, activating the corresponding pen (since there are two pens, the color circle is split into two ranges - 'warm' and 'cold'). With this process the collective set of robots generate compositions where from a random background some color clusters emerge.

RAP (Robotic Action Painter), 2006, work alone but based on the same principles of emergence and stigmergy. Some improvements however produce rather distinct compositions from those of the ArtSbot swarm. Some of the new skills are: to determined the length and shape of each trace, the capacity to decide, in a non-linear mode, the moment to stop and the ability to sign. Additionally RAP works with six color pens and the RGB sensors are disposed in a grid of 3x3 which permits to detect local patterns and not only colors.

If any of the action paintings done with the RAP are heavily formal, it is only a derivative form of it for the outcome often lacks the complexity or efficiency of the formalist stance at its highest (let alone the anachronism of it). Why Leonel Moura chooses the formalist paradigm out from many others available is another interesting question that is left unanswered in subsequent or previous work - why not a conceptual robot for instance? Besides, the robot is only but a poor surrogate of a formalist painter thus doing more harm to the Strong AI cause than the other way around. With the RAP, Leonel Moura risks being labeled a naturalist artist who attempts to mimic an image of another artist he would not dare to be, that of an Abstract Expressionist he finally fails to render and bring to life.

ISU, 2006, is very similar to RAP but is able to write letters and build words. In this fashion it makes compositions that resemble some of the Lettrist works from the 50’s and automatism.
Robotarium X is a large-scale steel glass construction lodging forty-five different robots, most powered by photovoltaic energy and fully autonomous.

Mark Pauline

Mark Pauline & son Jake

Mark Pauline (born December 14, 1953) is an American performance artist and inventor, best known as founder and director of Survival Research Labs.

Pauline founded SRL in 1978 and it is considered the premier practitioner of "industrial performing arts", and the forerunner of large scale machine performance. SRL is known for producing the most dangerous shows on earth. Although acknowledged as a major influence on popular competitions pitting remote-controlled robots and machines against each other, such as *BattleBots* and *Robot Wars*, Pauline shies away from rules-bound competition preferring a more anarchic approach. Machines are liberated and re-configured away from the functions they were originally meant to perform.

Pauline has written of SRL, "Since its inception SRL has operated as an organization of creative technicians dedicated to re-directing the techniques, tools, and tenets of industry, science, and the military away from their typical manifestations in practicality, product or warfare." Since its beginning through the end of 2006, SRL has conducted about 48 shows.

In August 1990, ArtPark, a state-sponsored arts festival in Lewiston, New York, cancelled a Pauline performance when it turned out he intended "to cover a sputtering Rube Goldberg spaceship with numerous Bibles" that would "serve as thermal protective shields" and be burned to ashes in the course of the performance.

According to Pauline "I like to make machines that can just do their own shows... machines that can do all that machines in the science fiction novels can do. I want to be there to make those dreams real."

Obtainium

The word obtainium likely originated from but was most certainly popularized by SRL crew finding or liberating discarded or obsolete items and re-directing them from industry, science, and the military and re-purposing them for anarchic machine performances.

Ken Rinaldo

Ken Rinaldo (born 1958) is an American artist and educator. He creates interactive art installations that explore the intersection between nature and technology. His robotic and bio-art installations seek to merge the organic and electromechanical seamlessly, expressing a gentle symbiosis.

His works are influenced by living systems theories, interspecies communication, artificial life research, and the idea of emergent properties. His work also deals with ecological issues often overlooked in favor of technological progress.

Ken Rinaldo's best known works are Autopoiesis (2000), an a-life robotic installation exploring the idea of group consciousness and Augmented Fish Reality (2004), a fish-driven robot .

Biography

He has an Associates in Science in Computer Science from Cañada College, 1982; a Bachelors of Art in Communications from The University of California, Santa Barbara; 1984 and a Masters in Fine Arts in Conceptual and Information Arts from San Francisco State University, 1996.

In 2000 he received the first prize at the VIDA 3.0 International Artificial Life Competition for Autopoiesis , in 2001 the same piece received an honorable mention at the Ars Electronica Festival. In 2004 Ken Rinaldo's art piece Augmented Fish Reality was awarded with an award of distinction at the same festival .

Rinaldo directs the Art & Technology program in the Department of Art at the main campus of the Ohio State University. He teaches interactive robotic sculpture, 3D modeling, rapid prototyping, motion graphics and animation.

Philosophy

Rinaldo is concerned with an idealized melding or an intersection that he believes is possible between natural and technological systems. He often asserts that integration of the natural and non-organic electro-mechanical elements are part of an important and very natural confluence and co-evolution that is necessary between living and our evolving technological material. His art works are influenced and evolve with research into living systems theory, artificial life and the current technologies we use to model and express mimesis through our current understanding of natural living systems.

Stelarc

Stelarc (Stelios Arkadiou) is a Greek-Australian performance artist whose works focuses heavily on extending the capabilities of the human body. As such, most of his pieces are centred around his concept that *the human body is obsolete*. Until 2007 he held the position of Principal Research Fellow in the Performance Arts Digital Research Unit at Nottingham Trent University in Nottingham, England. He is currently a visiting Professor in the School of Arts at Brunel University, West London.

Performances

Parasite: Event for Invaded and Involuntary Body, at the 1997 Ars Electronica Festival

Stelarc's idiosyncratic performances often involve robotics or other relatively modern technology integrated with his body somehow. In 25 different performances he has suspended himself in flesh hook suspension, often with one of his robotic inventions integrated. In another performance he allowed his body to be controlled remotely by electronic muscle stimulators connected to the internet. He has also performed with a robotic third arm, and a pneumatic spider-like six-legged walking machine which sits the user in the centre of the legs and allows them to control the machine through arm gestures. His works have been heralded for their abilities to embrace a wider audience, the best example of this was his allowance for the worldwide audience to log into the exhibition and thus access or control the electrodes his own body was hooked up to.

Third ear

In 2007, Stelarc had a cell-cultivated ear surgically attached to his left arm.

Works

In 2005, MIT Press published *Stelarc: The Monograph* which is the first extensive study of Stelarc's prolific work. It includes images of performances and interviews with several writers including William Gibson, who recount their meetings with Stelarc.

Awards and honors

- In 1995 Stelarc was awarded a three year fellowship from the Australia Council for the Arts.
- In 1997 Carnegie Mellon University appointed him Honorary Professor of Art and Robotics.
- In 1998 he was artist-in-residence for the city of Hamburg, Germany.
- In 2000 Monash University awarded him an Honorary Degree of Laws.
- In March, 2003, at Ohio State University, he completed an artist-in-residence program.
- In 2008 he was appointed as Senior Research Fellow and Artist-in-Residence, MARCS Auditory Laboratories, University of Western Sydney, Australia.
- In 2010 Stelarc received the prestigious Ars Electronica Golden Nica in the category "Hybrid Art", Linz, Austria.

Bill Vorn

Born in Montreal in 1959, **Bill Vorn** is a Canadian artist working in the field of Robotic art. His installation projects involve robotics and motion control, sound, lighting, video and cybernetic processes. He pursues research on Artificial Life and Agent technologies through artistic work based on the Aesthetics of Artificial Behaviors.

He holds a Ph.D. degree in Communication Studies from UQAM (Montreal) for a thesis on Artificial Life as a Media. He teaches Electronic Arts in the Department of Studio Arts at Concordia University (Intermedia/Cyberarts program) where he is responsible of the alab, a Robotic Art research lab part of the Hexagram Institute.

His work has been presented in many international events, including Ars Electronica,FILE Electronic Language International Festival , ISEA, DEAF, Sonar, Art Futura, EMAF and Artec. He has been awarded the Life 2.0 award (1999, Madrid), the Leprecon Award for Interactivity (1998, New York), the Prix Ars Electronica Distinction award (1996, Linz) and the International Digital Media Award (1996, Toronto). He has worked in collaboration with many Canadian artists (including Edouard Lock, Robert

Lepage, Gilles Maheu, LP Demers and Istvan Kantor). He was cofounder of the electronic music band Rational Youth with Tracy Howe in 1981.

Chico MacMurtrie

Chico MacMurtrie was born in New Mexico in 1961. He has been awarded four grants from the National Endowment for the Arts for Interdisciplinary Artists. In 1990 he received the San Francisco Bay Guardian Goldie Award.

In 1992 MacMurtrie formed Amorphic Robot Works, a group of artists and engineers working together to create robotic art performances and installations. His permanent commissioned interactive sculptures include the anthropomorphic *Urge to Stand* at Yerba Buena Gardens in San Francisco, *Fetus to Man* sculpture/clock for the city of Lille France, and *Growing Raining Tree* at Contemporary Arts Center in Cincinnati OH.

ARW's Artistic Director Chico MacMurtrie, describes his vision, "The work is an ongoing endeavor to uncover the primacy of movement and sound. Each machine is inspired or influenced, both, by modern society, and what I physically experience and sense. The whole of this input informs my ideas and work."

Flaming Lotus Girls

The San Francisco, California,-based **Flaming Lotus Girls** is a collaborative art group that creates large-scale fire art. The group was founded in 2000 by a group of artists working on a sculpture for the Burning Man arts festival in the Black Rock Desert in Nevada. They take their name from their first sculpture, *Flaming Lotus Sr*. Their work is notable for being interactive and kinetic. Their pieces are composed primarily of steel, stainless steel, copper, glass, wood, and light, with flames ranging from a 2-inch (51 mm) flicker to a 150-foot (46 m) blaze.

One distinctive element is that this is a female driven group. A core mission of the group is to help women learn how to use their hands and tools to build sculpture. While they invite both women and men, from expert to novice, to collaborate, all must consider themselves "girls".

Any member who regularly attends meetings can have input in designing, building, operating and playing with the large-scale work that is created. This type of leadership has been coined a *do-ocracy*. In short, if one wants to see something happen, they take the initiative to do it.

Art

Soma (2009)

Soma translates the anatomy of neurons into metal, fire and light; magnifying the microscopic world to an epic scale. In Soma, an elegant axon arch connects an earthbound neuron with its partner floating overhead. Fire and light flow like electrochemical signals between Soma's two neurons. Spinning balls of fire form the neuron's nuclei. Slender dendrites extend to the sky and reach down to the earth, emitting constant flame and color changing light. Computer-controlled flame and sequenced LEDS travel in patterns along the raised Soma archway, creating a spectacular, unique show. Throughout the sculpture, fire and LED lighting accentuate not only the curves and geometric forms of the sculpture, but also the very concept of neuronal transmission of information.

All of the fire effects on Soma are participant controlled, allowing the public to communicate with the sculpture itself. Soma offers participants an interactive installation that investigates the very basis of intellectual communication. It invites them to be an active force in this communication, providing input via the controls, sharing their knowledge and creating messages of warmth. A Soma is a cell body of a neuron with branching dendrites (signal receivers) and a projection called an axon, which conducts the nerve signal. At the other end of the axon, the axon terminals transmit electro-chemical signals across a synapse (the gap between the axon terminal and the receiving cell). Soma is an interactive sculptural installation depicting two communicating neurons made of stainless steel, copper, aluminum, bronze, resin, fire and light. Each of Soma's two neurons has a spinning fire nucleus. The nuclei are counter spinning balls of flame with variable speed motors.

Soma is 25 feet (7.6 m) high and 50 feet (15 m) long. It is roughly a rectangular shape that occupies approximately 5,000 square feet (460 m^2), including the fuel depot. She uses up to 100 gallons of fuel per hour. There are 35 Dendrites using approximately 21 feet (6.4 m) of stainless steel tubing each. 735 feet (224 m) of stainless steel tubing was used for dendrites over all. Two dodecahedrons constructed from 24 stainless steel pentagons comprise the cell bodies of Soma, and enclose the nuclei. Each pentagon used about 10 feet (3.0 m) of stainless steel tubing. A total 240 feet (73 m) of stainless steel tubing was used for the dodecahedrons. There are flame effects running down the axon which simulate signal neurotransmission. Participants control the "neurotransmission" by pushing buttons. A "Sparkle Poof" simulates release of neurotransmitters at the synapse. Each aerial dendrite and the axon burn with continuous flame effects. Soma has more than 60 custom LED packages, each housed in a unique casting constructed of resin and aluminum that can each output 270 lumens of light, each individually controllable and capable of 16 million colors.

Mutopia (2008)

Mutopia methanol shooters, Friday night, Burning Man 2008

Mutopia is a spiraling sculptural installation of Seedpods laid out according to the Golden Ratio, a proportion found throughout art and nature. Made of interactive fire, cast aluminum, LED light, video screens, sensors, steel, copper and stainless steel, these thirteen Seedpods lie within a 97-by-60-foot (30 × 18 m) footprint. Each Pod showcases a different stage of growth, from emergence to full bloom. The outer spiral initiates with a Seedpod just emerging from the surface of the playa, the newest germination of Mutopia's growth. Although small, it is the most evolved, exhibiting significant technological advancement and the greatest sensory awareness of the presence of other lifeforms. As one moves towards the center, one is transported back in Mutopia's evolutionary history: the Seedpods rise in height and slowly devolve into their more ancient forms.

During the day, the outgrowths and overhangs of Mutopia provide a still oasis, awaiting the dusk to open its blooms. As the day progresses and wind blows across the playa, something flashes in the sunlight amid the dust, drawing wanderers towards Mutopia. As they approach, they can hear sounds fluttering at the roots and swirling overhead under reflective leaves, beckoning them to come nearer. Mutopia rewards their curiosity with a reaction to their presence. Trying to communicate, the organic lifeforms reflect the participants' voices back to them and illuminate visual messages on integrated screens.

Further in the inner sanctum of the installation, citizens can manipulate the form of Mutopia by turning cranks and pulling levers.

At night Mutopia comes alive – illuminated with light and fire that beckon visitors to enter. The newest Pods are rumbling, hissing steam and glowing with an inner light which occurs with greater or lesser intensity according to human proximity. These Seedpods, aware of their environment and of human presence, respond with a flicker of video and visuals emanating from fruits that emerge from the cast aluminum Seedpods. Moving along the spiral arm, the Pods are extending roots, cracking open, exposing the beginnings of growth and a fire's glow. A delicate ambient flame quickly develops into propane-driven pneumatics, bursting with fire and movement. Moving on, growth is exponential, roots are solidified and the reproductive parts gain form and move with the force of participant-controlled multi-directional propane "poofers". Towards the inner vortex, the plants extend tendrils that intertwine, creating archways for participants to gaze through and walk beneath. Humans activate these Pods by using hand cranks and levers to articulate motion and their voices to direct the fire effects. In the inner sanctum of the spiral, the leaves have interwoven to form a canopy of multicolored computer-controlled fire.

In its final stages of growth, Mutopia's flowers bloom with a show of multi-colored, multi-directional computer-controlled, liquid-fueled fire. At the zenith of maturity, Mutopia majestically projects its seeds and pollen into the environment with a dramatic final blast from the inner sanctum. Completing the cycle, the new seeds of Mutopia start a purposeful journey to new environments where they will lie dormant until watered and cultivated by the creative juices of a new society. Portions of the incomplete sculpture were presented at Maker Faire 2008 and Yuri's Night at NASA Ames.

The Serpent Mother (2006)

The Serpent Mother

The Serpent Mother at Dawn 2006

The Serpent Mother is a 168-foot-long (51 m) sculpture of a skeletal serpent coiled around and protecting her egg. Propane fueled flame effects light up the top of each vertebra. Six-foot-high (1.8 m) jets of flame shoot from each vertebra when participants push buttons near the base of the sculpture or by computer control. The neck and jaw are operated by pneumatic cylinders. The top of the arch of her back is 20 feet (6.1 m) off the ground. Each of her teeth is lit from within by fire and her underbelly is lit by undulating computer controlled LEDs. The 10-foot-tall (3.0 m) egg has a finely worked copper shell. It opens and shoots colored liquid methanol flame effects. The whole piece burns approximately 2 tons of propane per evening.

Over a hundred people at a time can stand inside the area enclosed by her curving body, being kept warm and entertained by the fires burning around them.

The completed work has been displayed at

- Robodock Arts and Technology Festival, Amsterdam, Netherlands, September 2007
- The Crucible (arts education center) Fire Arts Festival, Oakland CA, July 2007
- Burning Man arts festival, Nevada, August 2006

- A partially completed version of the sculpture was displayed at The Crucible Fire Arts Festival, July 2006

The Angel of the Apocalypse (2005)

The Angel of the Apocalypse at Burning Man 2005

This elegant sculptural group, built of steel, driftwood and fire, rises from the earth in the form of an abstracted bird.

The Angel's wings burn continuously with ambient flame, each feather brought to life with blazing “poof” fire effects controlled by participants gathered in the courtyard of the 50 by 50-foot (15 m) installation. Its great head, formed of curved steel plate and enlivened with hand-blown glass eyes, stands 20 feet (6.1 m) tall and functions as a wood burning fireplace.

Visitors are invited to move around and between the Angel’s flaming feathers and to seat themselves atop its driftwood torso. All are encouraged to operate the propane flame effects of its inner wings.

The Seven Sisters (2004)

Electra, of the Seven Sisters morning after the Burn, Burning Man 2004

A collection of seven sculptures approximately 15 feet (4.6 m) in height representing the stars of the Pleiades constellation.

- Alcyone
- Celano
- Maia
- Taygeta
- Asterope
- Merope
- Electra

The Hand of God (2003)

A 12-foot (3.7 m) copper sculpture of a woman's hand that shoots flame from all five digits.

Mini Mega Jr. (2002)

Fire Fan (2002)

Huge plumes of liquid fire controlled by midi.

Fire Island (2002)

Interactive Flaming Flowers, cacti, arbors and more.

Fire Garden (2001)

A garden of fire, including copper flowers, a lily pond, and a weeping willow.

Flaming Lotus Sr. (2000)

A sculptural flame thrower.

Leonel Moura

Leonel Moura (born on December 26, 1948 in Lisbon, Portugal) is a conceptual artist whose work shifted in the late 90's from photo based work to Artificial Intelligence and Robotic art. Since then he has produced several Painting Robots and the Robotarium, a zoo for robots, the first of its kind in the world. RAP (Robotic Action Painter), 2006, a robot that makes drawings based on emergence and stigmergy; decides when the work is ready and signs it, is displayed as a permanent installation at the American Museum of Natural History in New York.

Robots

ArtSbot (Art Swarm Robots), 2003, comprise several small autonomous robots, called Mbots, each equipped with color detection sensors, obstacle avoidance sensors, a microcontroller and actuators, for locomotion and pen manipulation. Mbots have two distinct behaviors: the random behavior that initializes the process by activating a pen, based on a small probability (usually 2/256), whenever the color sensors read white; and the positive feed-back behavior that reinforces the color detected by the sensors, activating the corresponding pen (since there are two pens, the color circle is split into two ranges - 'warm' and 'cold'). With this process the collective set of robots generate compositions where from a random background some color clusters emerge.

RAP (Robotic Action Painter), 2006, work alone but based on the same principles of emergence and stigmergy. Some improvements however produce rather distinct compositions from those of the ArtSbot swarm. Some of the new skills are: to determined the length and shape of each trace, the capacity to decide, in a non-linear mode, the moment to stop and the ability to sign. Additionally RAP works with six color pens and the RGB sensors are disposed in a grid of 3x3 which permits to detect local patterns and not only colors.

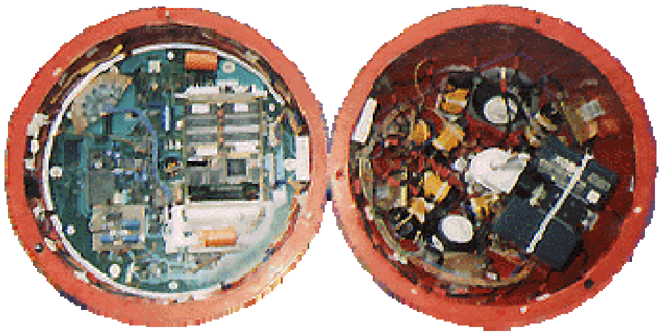
If any of the action paintings done with the RAP are heavily formal, it is only a derivative form of it for the outcome often lacks the complexity or efficiency of the formalist stance at its highest (let alone the anachronism of it). Why Leonel Moura chooses the formalist paradigm out from many others available is another interesting question that is left unanswered in subsequent or previous work - why not a conceptual robot for instance? Besides, the robot is only but a poor surrogate of a formalist painter thus doing more harm to the Strong AI cause than the other way around. With the RAP, Leonel Moura risks being labeled a naturalist artist who attempts to mimic an image of another artist he would not dare to be, that of an Abstract Expressionist he finally fails to render and bring to life.

ISU, 2006, is very similar to RAP but is able to write letters and build words. In this fashion it makes compositions that resemble some of the Lettrist works from the 50's and automatism.
Robotarium X is a large-scale steel glass construction lodging forty-five different robots, most powered by photovoltaic energy and fully autonomous.

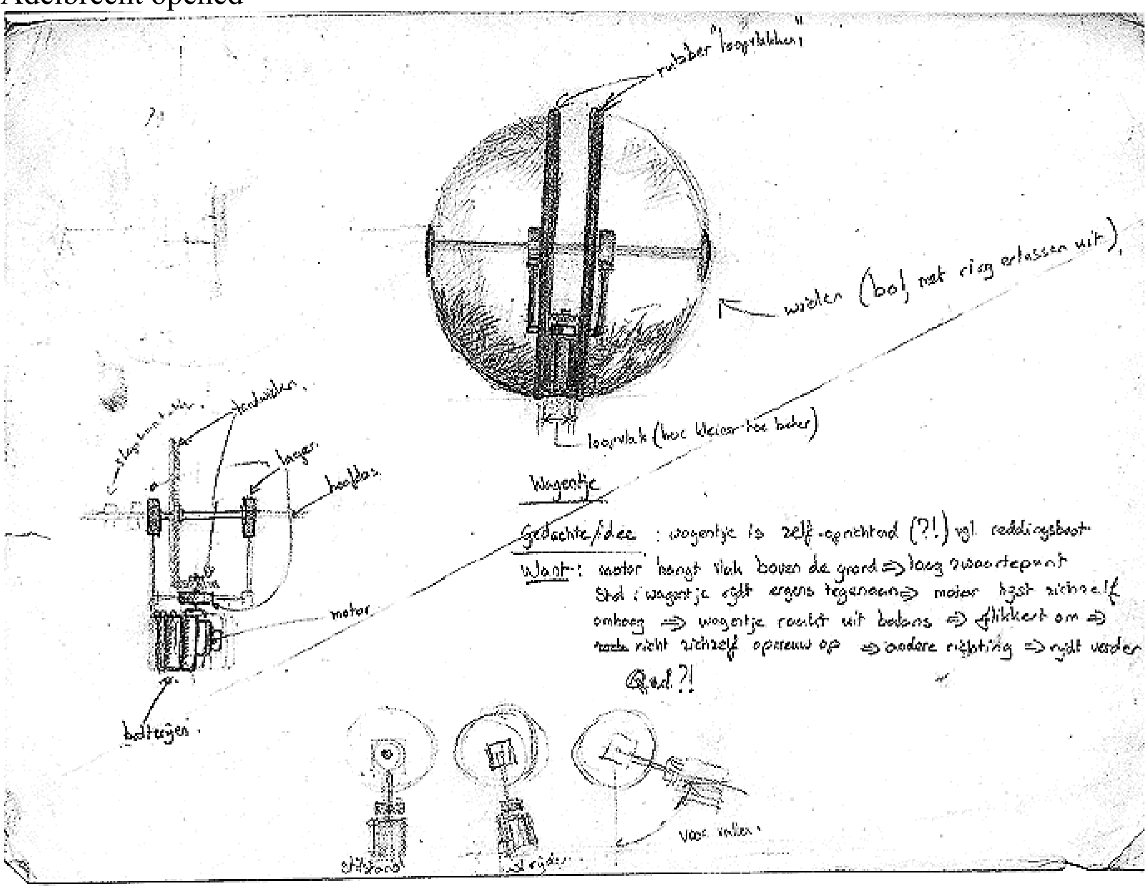
Adelbrecht

Adelbrecht first version

Adelbrecht was a speaking, interactive robot in the form of a ball, designed by **Martin Spanjaard** (born 1952 in Haarlem, Netherlands).
A first, simple version of Adelbrecht was presented in 1985.
Starting 1988 a second, more powerful version was developed. In 1992 it got an honourable mentioning at the Prix Ars Electronica. Its last performance, during newyears eve of 2000, never happened because of a serious hardware failure. Since then Martin Spanjaard proclaimed Adelbrecht a 'dead robot'.



Adelbrecht opened



First design sketch

Etymology

The name "Adelbrecht" is a name consisting of two Middle Dutch words, Adel (noble) and Brecht (bright). A modern version of this name is Albert or Elbert.

Chapter- 4

Robotic Architectures

ATLANTIS architecture

The **A Three-Layer Architecture for Navigating Through Intricate Situations (ATLANTIS)** is a hybrid reactive/deliberative robot architecture developed by Erann Gat at the Jet Propulsion Laboratory.

Agent architecture

Agent architecture in computer science is a blueprint for software agents and intelligent control systems, depicting the arrangement of components. The architectures implemented by intelligent agents are referred to as cognitive architectures.

Layered architectures include:

Subsumption architecture

Subsumption architecture is a reactive robot architecture heavily associated with behavior-based robotics. The term was introduced by Rodney Brooks and colleagues in 1986 . Subsumption has been widely influential in autonomous robotics and elsewhere in real-time AI.

Description

A subsumption architecture is a way of decomposing complicated intelligent behaviour into many "simple" behaviour modules, which are in turn organized into layers. Each layer implements a particular goal of the agent, and higher layers are increasingly abstract. Each layer's goal subsumes that of the underlying layers, e.g. the decision to move forward by the eat-food layer takes into account the decision of the lowest obstacle-avoidance layer. As opposed to more traditional AI approaches subsumption architecture uses a bottom-up design.

For example, a robot's lowest layer could be "avoid an object", on top of it would be the layer "wander around", which in turn lies under "explore the world". Each of these horizontal layers access all of the sensor data and generate actions for the actuators — the main caveat is that separate tasks can suppress (or overrule) inputs or inhibit outputs. This way, the lowest layers can work like fast-adapting mechanisms (e.g. reflexes), while the higher layers work to achieve the overall goal. Feedback is given mainly through the environment.

Attributes of the architecture

The main advantages of the methodology are:

- the modularity,
- the emphasis on iterative development & testing of real-time systems in their target domain, and
- the emphasis on connecting limited, task-specific perception directly to the expressed actions that require it.

These innovations allowed the development of the first robots capable of animal-like speeds.

The main disadvantages of this model are:

- the inability to have many layers, since the goals begin interfering with each other,
- the difficulty of designing action selection through highly distributed system of inhibition and suppression, and
- the consequent rather low flexibility at runtime.

Cognitive architecture

A **cognitive architecture** is a blueprint for intelligent agents. It proposes (artificial) computational processes that act like certain cognitive systems, most often, like a person, or acts intelligent under some definition. Cognitive architectures form a subset of general agent architectures. The term 'architecture' implies an approach that attempts to model not only behavior, but also structural properties of the modelled system. These need not be physical properties: they can be properties of virtual machines implemented in physical machines (e.g. brains or computers).

Characterization

Common among researchers on cognitive architectures is the belief that understanding (human, animal or machine) cognitive processes means being able to implement them in a working system, though opinions differ as to what form such a system can have: some researchers assume that it will necessarily be a symbolic computational system whereas others argue for alternative models such as connectionist systems or dynamical systems. Cognitive architectures can be characterized by certain properties or goals, as follows, though there is not general agreement on all aspects:

1. Implementation of not just various different aspects of cognitive behavior but of cognition as a whole (Holism, e.g. Unified theory of cognition). This is in contrast to cognitive models, which focus on a particular competence, such as a kind of problem solving or a kind of learning.
2. The architecture often tries to reproduce the behavior of the modelled system (human), in a way that timely behavior (reaction times) of the architecture and modelled cognitive systems can be compared in detail.
3. Robust behavior in the face of error, the unexpected, and the unknown.
4. Learning (not for all cognitive architectures)
5. Parameter-free: The system does not depend on parameter tuning (in contrast to Artificial neural networks) (not for all cognitive architectures)
6. Some early theories such as SOAR and ACT-R originally focused only on the 'internal' information processing of an intelligent agent, including tasks like reasoning, planning, solving problems, learning concepts. More recently many architectures (including SOAR, ACT-R, PreAct, ICARUS, CLARION), FORR have expanded to include perception, action and also affective states and processes including motivation, attitudes, and emotions.
7. On some theories the architecture may be composed of different kinds of sub-architectures (often described as 'layers' or 'levels') where the layers may be distinguished by types of function, types of mechanism and representation used, types of information manipulated, or possibly evolutionary origin. These are hybrid architectures (e.g., CLARION).
8. Some theories allow different architectural components to be active concurrently, whereas others assume a switching mechanism that selects one component or module at a time, depending on the current task. Concurrency is normally required for an architecture for an animal or robot that has multiple sensors and effectors in a complex and dynamic environment, but not in all robotic paradigms.
9. Most theories assume that an architecture is fixed and only the information stored in various subsystems can change over time (e.g. Langley et al., below), whereas others allow architectures to grow, e.g. by acquiring new subsystems or new links between subsystems (e.g. Minsky and Sloman, below).

It is important to note that cognitive architectures don't have to follow a top-down approach to cognition (cf. Top-down and bottom-up design).

Distinctions

Cognitive architectures can be symbolic, connectionist, or hybrid. Some cognitive architectures or models are based on a set of generic rules, as, e.g., the Information Processing Language (e.g., Soar based on the unified theory of cognition, or similarly ACT). Many of these architectures are based on the-mind-is-like-a-computer analogy. In contrast subsymbolic processing specifies no such rules a priori and relies on emergent properties of processing units (e.g. nodes). Hybrid architectures combine both types of processing (such as CLARION). A further distinction is whether the architecture is centralized with a neural correlate of a processor at its core, or decentralized (distributed). The decentralized flavor, has become popular under the name of parallel distributed processing in mid-1980s and connectionism, a prime example being neural networks. A further design issue is additionally a decision between holistic and atomistic, or (more concrete) modular structure. By analogy, this extends to issues of knowledge representation.

In traditional AI, intelligence is often programmed from above: the programmer is the creator, and makes something and imbues it with its intelligence, though many traditional AI systems were also designed to learn (e.g. improving their game-playing or problem-solving competence). Biologically inspired computing, on the other hand, takes sometimes a more bottom-up, decentralised approach; bio-inspired techniques often involve the method of specifying a set of simple generic rules or a set of simple nodes, from the interaction of which emerges the overall behavior. It is hoped to build up complexity until the end result is something markedly complex. However, it is also arguable that systems designed top-down on the basis of observations of what humans and other animals can do rather than on observations of brain mechanisms, are also biologically inspired, though in a different way.

Cognitive Architecture Include:-

Soar (cognitive architecture)

Soar is a symbolic cognitive architecture, created by John Laird, Allen Newell, and Paul Rosenbloom at Carnegie Mellon University, now maintained by John Laird's research group at the University of Michigan. It is both a view of what cognition is and an implementation of that view through a computer programming architecture for Artificial Intelligence (AI). Since its beginnings in 1983 and its presentation in a paper in 1987, it has been widely used by AI researchers to model different aspects of human behavior.

Theory

The main goal of the Soar project is to be able to handle the full range of capabilities of an intelligent agent, from highly routine to extremely difficult open-ended problems. In order for that to happen, according to the view underlying Soar, it needs to be able to create representations and use appropriate forms of knowledge (such as procedural, declarative, episodic). Soar should then address a collection of mechanisms of the mind. Also underlying the Soar architecture is the view that a symbolic system is essential for general intelligence. This is known as the *physical symbol system hypothesis*. The views of cognition underlying Soar are tied to the psychological theory expressed in Allen Newell's book, *Unified Theories of Cognition*.

While symbol processing remains the core mechanism in the architecture, recent versions of the theory incorporate non-symbolic representations and processes, including reinforcement learning, imagery processing, and emotion modeling (Laird, 2008).

Although the ultimate goal for Soar is to achieve general intelligence, there is no claim that this goal has already been reached. Advocates of the system recognize that Soar is still missing some important aspects of intelligence. Some examples of missing capabilities include automatically creating new representations on its own, such as through hierarchical clustering.

Architecture

Soar is based on a production system, i.e. it uses explicit production rules to govern its behavior (these are roughly of the form "if... then...", as also used in expert systems). Problem solving can be roughly described as a search through a *problem space* (the collection of different states which can be reached by the system at a particular time) for a *goal state* (which represents the solution for the problem). This is implemented by searching for the states which bring the system gradually closer to its goal. Each move consists of a decision cycle which has an elaboration phase (in which a variety of different pieces of knowledge bearing the problem are brought to Soar's working memory) and a decision procedure (which weighs what was found on the previous phase and assigns preferences to ultimately decide the action to be taken). In addition to problem space search, however, Soar can be used to instantiate reasoning techniques such as reinforcement learning which do not require detailed internal models of the environment. In this way, Soar is flexible to behaving when varying amounts of task knowledge are available.

SOAR originally stood for *S*tate, *O*perator *A*nd *R*esult, reflecting this representation of problem solving as the application of an operator to a state to get a result. According to the project FAQ, the Soar development community no longer regards Soar as an acronym so it is no longer spelled all in caps though it is still representative of the core of the implementation.

If the decision procedure just described is not able to determine a unique course of action, Soar may use different strategies, known as *weak methods* to solve the impasse. These methods are appropriate to situations in which knowledge is not abundant. Some examples are *means-ends analysis* (which may calculate the difference between each available option and the goal state) and a type of hill-climbing. When a solution is found by one of these methods, Soar uses a learning technique called *chunking* to transform the course of action taken into a new rule. The new rule can then be applied whenever Soar encounters the situation again (that is, there will no longer be an impasse).

ACT-R is another cognitive architecture by John R. Anderson that operates on similar principles. Other cognitive architectures are CHREST, CLARION, ICARUS, DUAL, and Psi.

ACT-R

ACT-R (pronounced *act-ARE*: Adaptive Control of Thought--Rational) is a cognitive architecture mainly developed by John Robert Anderson at Carnegie Mellon University. Like any cognitive architecture, ACT-R aims to define the basic and irreducible cognitive and perceptual operations that enable the human mind. In theory, each task that humans can perform should consist of a series of these discrete operations.

Most of the ACT-R basic assumptions are also inspired by the progresses of cognitive neuroscience, and ACT-R can be seen and described as a way of specifying how the brain itself is organized in a way that enables individual processing modules to produce cognition.

Inspiration

ACT-R has been inspired by the work of Allen Newell, and especially by his life-long championing the idea of unified theories as the only way to truly uncover the underpinnings of cognition . In fact, John Anderson usually credits Allen Newell as the major source of influence over his own theory.

What ACT-R looks like

Like other influential cognitive architectures (including Soar, CLARION, and EPIC), the ACT-R theory has a computational implementation as an interpreter of a special coding language. The interpreter itself is written in Lisp, and might be loaded into any of the most common distributions of the Lisp language.

This means that any researcher may download the ACT-R code from the ACT-R website, load it into a Lisp distribution, and gain full access to the theory in the form of the ACT-R interpreter.

Also, this enables researchers to specify models of human cognition in the form of a script in the ACT-R language. The language primitives and data-types are designed to reflect the theoretical assumptions about human cognition. These assumptions are based on numerous facts derived from experiments in cognitive psychology and brain imaging.

Like a programming language, ACT-R is a framework: for different tasks (e.g., Tower of Hanoi, memory for text or for list of words, language comprehension, communication, aircraft controlling), researchers create "models" (i.e., programs) in ACT-R. These models reflect the modelers' assumptions about the task within the ACT-R view of cognition. The model might then be run.

Running a model automatically produces a step-by-step simulation of human behavior which specifies each individual cognitive operation (i.e., memory encoding and retrieval, visual and auditory encoding, motor programming and execution, mental imagery manipulation). Each step is associated with quantitative predictions of latencies and accuracies. The model can be tested by comparing its results with the data collected in behavioral experiments.

In recent years, ACT-R has also been extended to make quantitative predictions of patterns of activation in the brain, as detected in experiments with fMRI. In particular, ACT-R has been augmented to predict the shape and time-course of the BOLD response of several brain areas, including the hand and mouth areas in the motor cortex, the left prefrontal cortex, the anterior cingulate cortex, and the basal ganglia.

Brief outline

ACT-R's most important assumption is that human knowledge can be divided into two irreducible kinds of representations: *declarative* and *procedural*.

Within the ACT-R code, declarative knowledge is represented in form of *chunks*, i.e. vector representations of individual properties, each of them accessible from a labelled slot.

Chunks are held and made accessible through *buffers*, which are the front-end of what are *modules*, i.e. specialized and largely independent brain structures.

There are two types of modules:

- **Perceptual-motor modules**, which take care of the interface with the real world (i.e., with a simulation of the real world). The most well-developed perceptual-motor modules in ACT-R are the visual and the manual modules.
- **Memory modules**. There are two kinds of memory modules in ACT-R:
    - **Declarative memory**, consisting of facts such as *Washington, D.C. is the capital of United States*, *France is a country in Europe*, or *2+3=5*
    - **Procedural memory**, made of productions. Productions represent knowledge about how we do things: for instance, knowledge about how to type the letter "Q" on a keyboard, about how to drive, or about how to perform addition.

All the modules can only be accessed through their buffers. The contents of the buffers at a given moment in time represents the state of ACT-R at that moment. The only exception to this rule is the procedural module, which stores and applies procedural knowledge. It does not have an accessible buffer and is actually used to access other module's contents.

Procedural knowledge is represented in form of *productions*. The term "production" reflects the actual implementation of ACT-R as a production system, but, in fact, a production is mainly a formal notation to specify the information flow from cortical areas (i.e. the buffers) to the basal ganglia, and back to the cortex.

At each moment, an internal pattern matcher searches for a production that matches the current state of the buffers. Only one such production can be executed at a given moment. That production, when executed, can modify the buffers and thus change the state of the system. Thus, in ACT-R, cognition unfolds as a succession of production firings.

The symbolic vs. connectionist debate

In the cognitive sciences, different theories are usually ascribed to either the "symbolic" or the "connectionist" approach to cognition. ACT-R clearly belongs to the "symbolic" field and is classified as such in standard textbooks and collections . Its entities (chunks and productions) are discrete and its operations are syntactical, that is, not referring to the semantic content of the representations but only to their properties that deem them appropriate to participate in the computation(s). This is seen clearly in the chunk slots and in the properties of buffer matching in productions, both of which function as standard symbolic variables.

Members of the ACT-R community, including its developers, prefer to think of ACT-R as a general framework that specifies how the brain is organized, and how its organization gives birth to what is perceived (and, in cognitive psychology, investigated) as mind, going beyond the traditional symbolic/connectionist debate. None of this, naturally, argues against the classification of ACT-R as symbolic system, because all symbolic approaches to cognition aim to describe the mind, as a product of brain function, using a certain class of entities and systems to achieve that goal.

A common misunderstanding suggests that ACT-R may not be a symbolic system because it attempts to characterize brain function. This is incorrect on two counts: First, because all approaches to computational modeling of cognition, symbolic or otherwise, must in some respect characterize brain function, because the mind is brain function. And second, because all such approaches, including connectionist approaches, attempt to characterize the mind at a cognitive level of description and not at the neural level, because it is only at the cognitive level that important generalizations can be retained.

Further misunderstandings arise because of the associative character of certain ACT-R properties, such as chunks spreading activation to each other, or chunks and productions carrying quantitative properties relevant to their selection. None of these properties counter the fundamental nature of these entities as symbolic, regardless of their role in unit selection and, ultimately, in computation.

Theory vs. implementation, and Vanilla ACT-R

The importance of distinguishing between the theory itself and its implementation is usually highlighted by ACT-R developers.

In fact, much of the implementation does not reflect the theory. For instance, the actual implementation makes use of additional 'modules' that exist only for purely computational reasons, and are not supposed to reflect anything in the brain (e.g., one computational module contains the pseudo-random number generator used to produce noisy parameters, while another holds naming routines for generating data structures accessible through variable names).

Also, the actual implementation is designed to enable researchers to modify the theory, e.g. by altering the standard parameters, or creating new modules, or partially modifying the behavior of the existing ones.

Finally, while Anderson's laboratory at CMU maintains and releases the official ACT-R code, other alternative implementations of the theory have been made available. These alternative implementations include *jACT-R* (written in Java by Anthony M. Harrison at the University of Pittsburgh) and *Python ACT-R* (written in Python by Terrence C. Stewart and Robert L. West at Carleton University, Canada).

Similarly, ACT-RN (now discontinued) was a full-fledged neural implementation of the 1993 version of the theory . All of these versions were fully functional, and models have been written and run with all of them.

Because of these implementational degrees of freedom, the ACT-R community usually refers to the "official", lisp-based, version of the theory, when adopted in its original form and left unmodified, as "Vanilla ACT-R".

Applications

Over the years, ACT-R models have been used in more than 700 different scientific publications, and have been cited in many more.

Memory, attention, and executive control

The ACT-R declarative memory system has been used to model human memory since its inception. In the course of years, it has been adopted to successfully model a large number of known effects. They include the fan effect of interference for associated information , primacy and recency effects for list memory , and serial recall .

ACT-R has been used to model attentive and control processes in a number of cognitive paradigms. These include the Stroop task , task switching , the psychological refractory period, and multi-tasking .

Natural language

A number of researchers have been using ACT-R to model several aspects of natural language understanding and production. They include models of syntactic parsing , language understanding, , language acquisition and metaphor comprehension .

Complex tasks

ACT-R has been used to capture how humans solve complex problems like the Tower of Hanoi , or how people solve algebraic equations . It has also been used to model human behavior in driving and flying.

With the integration of perceptual-motor capabilities, ACT-R has become increasingly popular as a modeling tool in human factors and human-computer interaction. In this domain, it has been adopted to model driving behavior under different conditions , menu selection and visual search on computer application , and web navigation .

Cognitive neuroscience

More recently, ACT-R has been used to predict patterns of brain activation during imaging experiments . In this field, ACT-R models have been successfully used to predict prefrontal and parietal activity in memory retrieval , anterior cingulate activity for control operations , and practice-related changes in brain activity .

Education

ACT-R has been often adopted as the foundation for cognitive tutors . These systems use an internal ACT-R model to mimic the behavior of a student and personalize his/her instructions and curriculum, trying to "guess" the difficulties that students may have and provide focused help.

Such "Cognitive Tutors" are being used as a platform for research on learning and cognitive modeling as part of the Pittsburgh Science of Learning Center. Some of the most successful applications, like the Cognitive Tutor for Mathematics, are used in thousands of schools across the United States.

Brief history

Early years: 1973-1990

ACT-R is the ultimate successor of a series of increasingly precise models of human cognition developed by John R. Anderson.

Its roots can be backtraced to the original HAM (Human Associative Memory) model of memory, described by John R. Anderson and Gordon Bower in 1973 . The HAM model was later expanded into the first version of the ACT theory . This was the first time the procedural memory was added to the original declarative memory system, introducing a computational dichotomy that was later proved to hold in human brain . The theory was then further extended into the ACT* model of human cognition .

Integration with rational analysis: 1990-1998

In the late eighties, Anderson devoted himself to exploring and outlining a mathematical approach to cognition that he named Rational Analysis . The basic assumption of Rational Analysis is that cognition is optimally adaptive, and precise estimates of cognitive functions mirror statistical properties of the environment . Later on, he came back to the development of the ACT theory, using the Rational Analysis as a unifying framework for the underlying calculations. To highlight the importance of the new approach in the shaping of the architecture, its name was modified to ACT-R, with the "R" standing for "Rational"

In 1993, Anderson met with Christian Lebiere, a researcher in connectionist models mostly famous for developing with Scott Fahlman the Cascade Correlation learning algorithm. Their joint work culminated in the release of ACT-R 4.0 . Thanks to Mike Byrne (now at Rice University), version 4.0 also included optional perceptual and motor capabilities, mostly inspired from the EPIC architecture, which greatly expanded the possible applications of the theory.

Current developments 1998-present

After the release of ACT-R 4.0, John Anderson became more and more interested in the underlying neural plausibility of his life-time theory, and began to use brain imaging techniques pursuing his own goal of understanding the computational underpinnings of human mind.

The necessity of accounting for brain localization pushed for a major revision of the theory. ACT-R 5.0 introduced the concept of modules, specialized sets of procedural and declarative representations that could be mapped to known brain systems . In addition, the interaction between procedural and declarative knowledge was mediated by newly introduced buffers, specialized structures for holding temporarily active information. Buffers were thought to reflect cortical activity, and a subsequent series of studies later confirmed that activations in cortical regions could be successfully related to computational operations over buffers.

A new version of the code, completely rewritten, was presented in 2005 as ACT-R 6.0. It also included significant improvements in the ACT-R coding language.

Spin Offs

The long development of the ACT-R theory gave birth to a certain number of parallel and related projects.

The most important ones are the PUPS production system, an initial implementation of Anderson's theory, later abandoned; and ACT-RN , a neural network implementation of the theory developed by Christian Lebiere.

Lynne Reder, also at Carnegie Mellon University, developed in the early nineties SAC, a model of conceptual and perceptual aspects of memory that shares many features with the ACT-R core declarative system, although differing in some assumptions.

LIDA (cognitive architecture)

The **LIDA** cognitive architecture is an integrated artificial cognitive system that attempts to model a broad spectrum of cognition in biological systems, from low-level perception/action to high-level reasoning. Developed primarily by Stan Franklin and colleagues at the University of Memphis, the LIDA architecture is empirically grounded in cognitive science and cognitive neuroscience. In addition to providing hypotheses to guide further research, the architecture can support control structures for software agents and robots. Providing plausible explanations for many cognitive processes, the LIDA conceptual model is also intended as a tool with which to think about how minds work.

Two hypotheses underlie the LIDA architecture and its corresponding conceptual model: 1) Much of human cognition functions by means of frequently iterated (~10 Hz) interactions, called cognitive cycles, between conscious contents, the various memory systems and action selection. 2) These cognitive cycles, serve as the "atoms" of cognition of which higher-level cognitive processes are composed.

Overview

Though it is neither symbolic nor strictly connectionist, LIDA is a hybrid architecture in that it employs a variety of computational mechanisms, chosen for their psychological plausibility. The LIDA cognitive cycle is composed of modules and processes employing these mechanisms.

Computational mechanisms

The LIDA architecture employs several modules that are designed using computational mechanisms drawn from the "new AI." These include variants of the Copycat Architecture , Sparse Distributed Memory , the Schema Mechanism , the Behavior Net , and the Subsumption Architecture.

Psychological and neurobiological underpinnings

As a comprehensive, conceptual and computational cognitive architecture the LIDA architecture is intended to model a large portion of human cognition . Comprising a broad array of cognitive modules and processes, the LIDA architecture attempts to implement and flesh out a number of psychological and neuropsychological theories including Global Workspace Theory , Situated Cognition , perceptual symbol systems , Working Memory , memory by affordances , long-term working memory , and the H-CogAff architecture .

LIDA’s cognitive cycle

The LIDA cognitive cycle can be subdivided into three phases, the understanding phase, the attention (consciousness) phase, and the action selection and learning phase. Beginning the understanding phase, incoming stimuli activate low-level feature detectors in Sensory Memory. The output engages Perceptual Associative Memory where higher-level feature detectors feed in to more abstract entities such as objects, categories, actions, events, etc. The resulting percept moves to the Workspace where it cues both Transient Episodic Memory and Declarative Memory producing local associations. These local associations are combined with the percept to generate a current situational model; the agent’s understanding of what is going on right now. The attention phase begins with the forming of coalitions of the most salient portions of the current situational model, which then compete for attention, that is a place in the current conscious contents. These conscious contents are then broadcast globally, initiating the learning and action selection phase. New entities and associations, and the reinforcement of old ones, occur as the conscious broadcast reaches the various forms of memory, perceptual, episodic and procedural. In parallel with all this learning, and using the conscious contents, possible action schemes are instantiated from Procedural Memory and sent to Action Selection, where they competes to be the behavior selected for this cognitive cycle. The selected behavior triggers Sensory-Motor Memory to produce a suitable algorithm for its execution, which completes the cognitive cycle.

History

Virtual Mattie (V-Mattie) is a software agent that gathers information from seminar organizers, composes announcements of next week's seminars, and mails them each week to a list that it keeps updated, all without the supervision of a human . V-Mattie employed many of the computational mechanisms mentioned above.

Baars’ Global Workspace Theory (GWT) inspired the transformation of V-Mattie into Conscious Mattie, a software agent with the same domain and tasks whose architecture included a consciousness mechanism à la GWT. Conscious Mattie was the first functionally, though not phenomenally, conscious software agent. Conscious Mattie gave rise to IDA.

IDA (Intelligent Distribution Agent) was developed for the US Navy to fulfill tasks performed by human resource personnel called detailers. At the end of each sailor's tour of duty, he or she is assigned to a new billet. This assignment process is called distribution. The Navy employs almost 300 full time detailers to effect these new assignments. IDA's task is to facilitate this process, by automating the role of detailer.

IDA was tested by former detailers and accepted by the Navy. Various Navy agencies supported the IDA project to the tune of some $1,500,000.

The LIDA (Learning IDA) architecture was spawned from IDA by the addition of several styles and modes of learning , as well as much else.

PreAct

PreAct is a cognitive engine technology that is driven both by changes in the state of the world and the intent of the system's users called associate systems. Associate systems are a knowledge-based system that are designed to work in conjunction with a human operator. An associate system observes the same data about the state of the world as the human operator, combining the information together to reach more abstract and aggregate conclusions about the state of the world. An associate system also observes the actions undertaken by a human operator, combining those actions with the state of the world to determine the operator's current objectives and activities. Based on the assessment of the state of the world and the activities and objectives of the human operator, the system can, within the bounds of its authority, carry out activities on behalf of the user, make the user aware of events particularly relevant to his activities, and manage the information content of the user's displays. Associate systems stand in contrast to autonomous systems, in which the software replaces the human operator, and decision-aiding systems, in which the software typically assists the human only when requested. An associate system acts as a teammate to a human operator; it can act without explicitly being ordered while remaining within the bounds of its authority, without taking final control away from the human operator.

Cougaar

Cougaar is a Java-based architecture for the construction of large-scale distributed multi-agent systems. Cougaar is the product of multi-year DARPA research projects ALP and Ultra*Log.

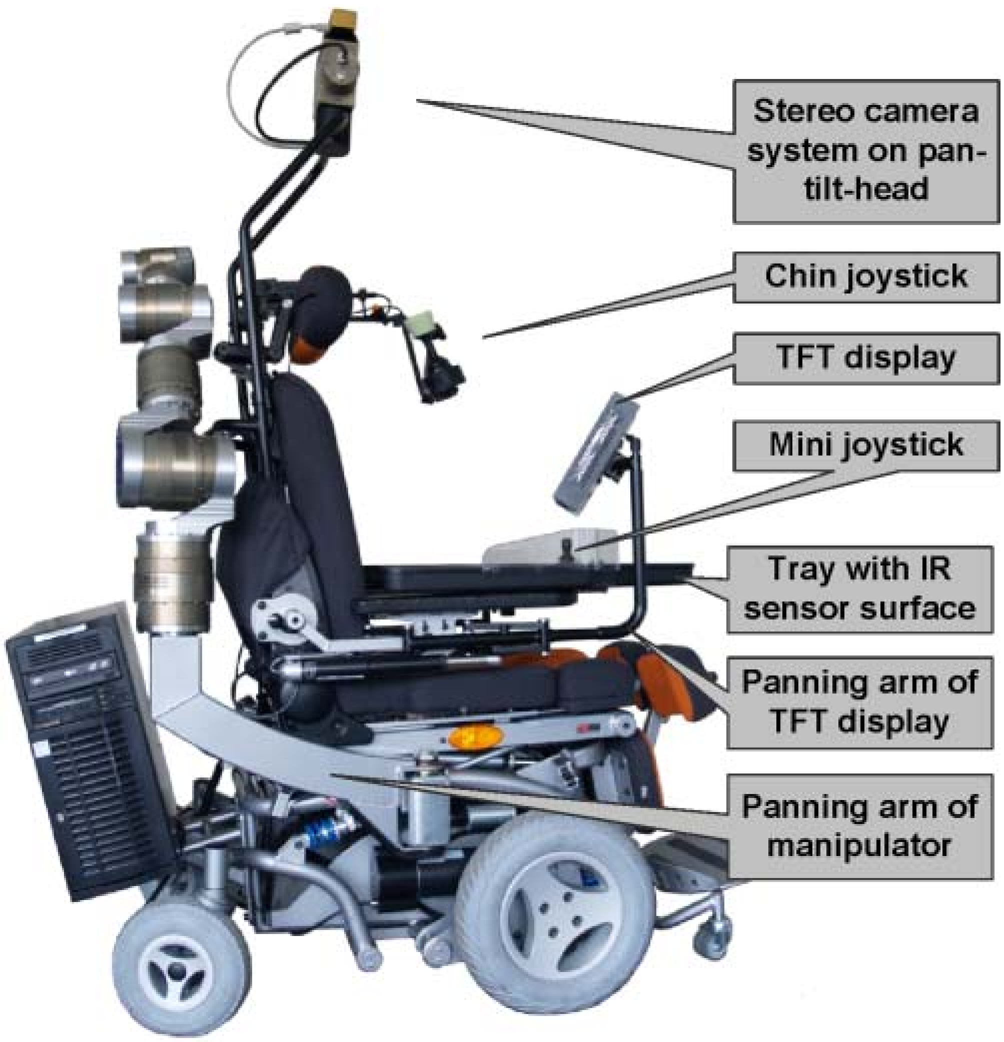
Cougaar is open source and includes not only the core runtime framework, but also a variety of demonstration, visualization, and management components to simplify the development of complex, distributed applications.

Cougaar is an abbreviation, of sorts, for "Cognitive Agent Architecture".

Autonomous robot architecture (AuRA)

The **Autonomous Robot Architecture (AuRA)** is a hybrid deliberative/reactive robot architecture developed by Ronald C. Arkin at the Georgia Institute of Technology.

Care-Providing Robot FRIEND



The Care-Providing Robot FRIEND

AMAROB logo

The care-providing robotic system FRIEND (*Functional Robot arm with user-frIENdly interface for Disabled people*) is a semi-autonomous robot designed to support disabled and elderly people in their daily life activities, like preparing and serving a meal, or reintegration in professional life. FRIEND make it possible for such people, e.g. patients which are paraplegic, have muscle diseases or serious paralysis, e.g. due to strokes, to perform special tasks in daily life self-determined and without help from other people like therapists or nursing staff.

The robot FRIEND is the third generation of such robots developed at the Institute of Automation (IAT) of University of Bremen within different research projects . Within the last project AMaRob, an interdisciplinary consortium, consisting of technicians, designers as well as therapists and further representatives of various interest groups, influences the development of FRIEND. Besides covering the various technical aspects, also design aspects were included as well as requirements from daily practice given by therapists, in order to develop a care-providing robot that is suitable for daily life activities. The AMaRob project was founded by the German Federal Ministry of Education and Research ("BMBF - Bundesministerium für Bildung und Forschung") within the "Leitinnovation Servicerobotik".

FRIEND::System

FRIEND is built from reliable industrial components. It is based on the wheelchair platform Nemo Vertical, which is an electrical wheelchair from Meyra. This basic platform has been equipped with various additional components, which are described in the following.

- **Robot Arm / Manipulator:** The Light Weight Robotic Arm 3 (LWA3) is a 7 degrees of freedom manipulator of Schunk mounted on an automated panning arm. So the arm can park behind the seat in order to navigate FRIEND in narrow passages. The robot arm is equipped with the prosthetic hand "SensorHand Speed" from Otto Bock which has built-in slip sensors in order to detect the slipping of gripped object and adapt the force accordingly. At the robot's wrist a force-torque sensor is mounted to perform force-torque-based reactive manipulative operations and to detect collisions.

- **TFT-Display:** The TFT display provides visual information to the user and is also mounted on a panning arm.

- **Intelligent Tray:** In front of the user an intelligent tray is available on which objects can be placed down by the manipulator. This tray is based on infra-red (IR) devices to acquire precise information about object locations, which should be manipulated.

- **Stereo Camera System:** A Bumblebee 2 stereo camera system with built-in calibration, synchronization and stereo projective calculation features is used to acquire information of the environment. It is mounted at the top of the system on a pan-tilt-head unit, which itself is installed on a special rack behind the seat.

- **Computer System:** A high-end PC unit is mounted on the wheelchair platform behind the user. The mounting as depicted is still in a prototype state.

- **Input devices:** There are several input devices which are available for FRIEND or under development: chin joystick, hand joystick, speech control (in- and output), brain-computer interface (BCI) and eye control. The input devices are adapted according to the impairments of the user or his preferences.

- **Infra-red Communication and Appliances:** An infra-red control unit, development by IGEL , for communication with various appliances in the robot's environment is integrated underneath the pan-tilt-head unit. Thus, e.g. an automatic door opening mechanism in the refrigerator and the microwave, the configuration and control of the microwave itself or various consumer electronic components can be operated wireless.

FRIEND::Scenarios

Within the AMaRob (AMaRob web page) project three scenarios were developed that support disabled and elderly people in their Activities of Daily Life (ADL) as well as in professional life.

ADL

This scenario enables the user to prepare and eat a meal. A special meal-tray has been designed which can be gripped by the manipulator. First the meal-tray is fetched from a refrigerator which is equipped with an automatic door opener. Then the manipulator puts the meal-tray in a remote controlled microwave oven to cook the meal. After cooking the meal-tray is fetched from the microwave oven and placed on the tray of the wheelchair. After that FRIEND supports the user to eat the meal. With a special designed spoon the manipulator can take a bit meal and feed the user. After the eating procedure the meal-tray is put away. The handles are designed in such a way that they can be robustly recognized by the vision system and easily grasped by manipulator's gripper.

Library

The first professional scenario is situated at a library service desk. Professional scenarios are very important for disabled people from the viewpoint of re-integration into daily life activities. With FRIEND the user can manage tasks like lending and return of books.

Workshop

The second professional scenario takes place in a rehabilitation workshop. The realized scenario is representative for many quality control tasks in industry and consists of checking the functionality of keypads for public telephone boxes. A keypad has to be taken from a keypad magazine by the gripper and put into a test adapter to check the correct working of pad's electronics. After that each keypad has to be inspected visually by the user to detect cracks or similar damages. Based on the results the keypad have to be sorted.

FRIEND::Architecture - MASSiVE

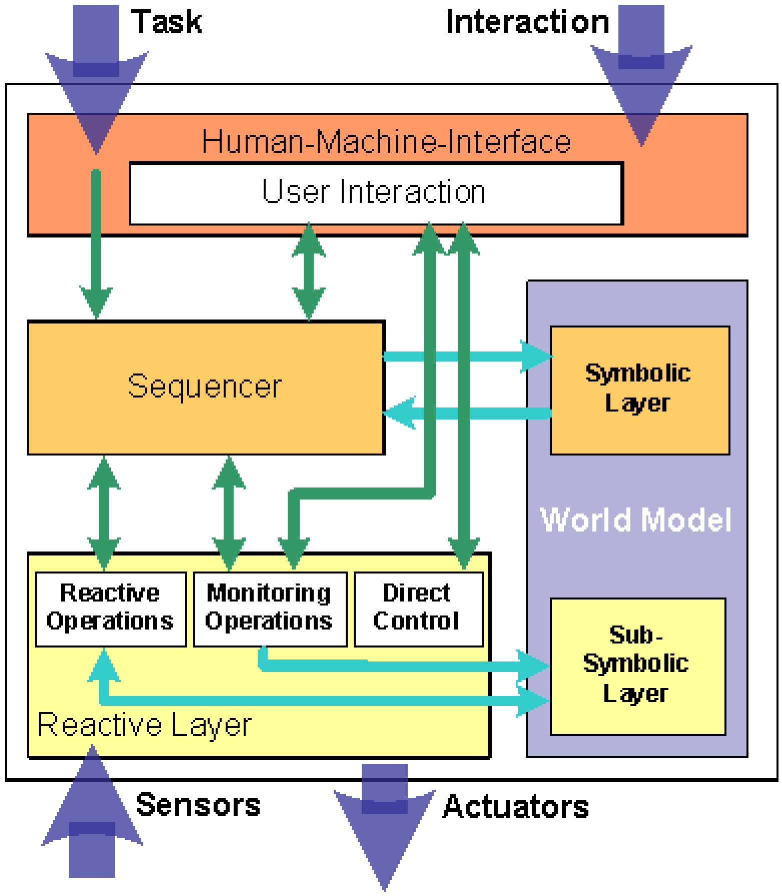


Fig. 2: Hybrid multi-layer control architecture for semi-autonomous service robots with verified task execution.

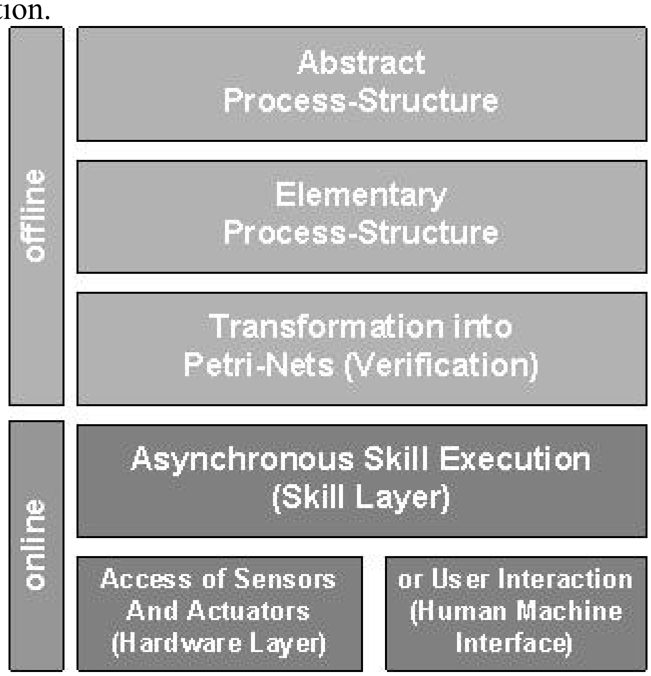


Fig. 3: Task knowledge specification, verification and execution concept in MASSiVE.

Fig. 4: Human Machine Interface (HMI) of care-providing robot FRIEND.

The realization of semi-autonomy is based on the central idea to include human tasks into task execution. A good explanation of this principle is given in the following quotation:

> *"When we think of interfaces between human beings and computers, we usually assume that the human being is the one requesting that a task be completed, and the computer is completing the task and providing the results. What if this process were reversed and a computer program could ask a human being to perform a task and return the results?*

With respect to a semi-autonomous service robotic system this means that the user's cognitive capabilities are taken into account, whenever a robust and reliable technical solution is unavailable. It is obvious that the acceptance of such a system will be low in general. But for people that are dependent on a personnel assistant, like the disabled or elderly, this approach offers the opportunity to decrease this dependency and therefore to increase their quality of life. A service robot has to be able to pursue a certain mission goal as commanded from its user but also needs to react flexibly to dynamic changes within the workspace. To meet these requirements, hybrid multi-layer control architectures have been successfully applied . These architectures usually consist of three layers:

- A *deliberative layer*, which contains a task planner to generate a sequence of operations to reach a certain goal with respect to the user's input command.
- A *reactive layer*, which has access to the system's sensors and actuators and provides reactive behavior which is robust even under environmental disturbances, for example with the help of closed-loop control.
- A *sequencer* that mediates between deliberator and reactive layer i.e. activates or deactivates reactive operations according to the deliberator's specification.

The software framework MASSiVE (*Multi-layer Architecture for Semi-autonomous Service robots with Verified task Execution*) is a special kind of hybrid multi-layer control architecture which is tailored to the requirements of semi-autonomous and distributed systems, like the care-providing robot FRIEND, acting in environments with distributed smart components. These intelligent wheelchair mounted manipulator systems allow to benefit from the inclusion of the user's cognitive capabilities into task execution and consequently lower the system complexity compared to a fully autonomous system. The semi-autonomous control requires a sophisticated integration of a human-machine-interface (HMI) which is able to couple input devices according to the user's impairment , for example a haptic suit, eye-mouse, speech-recognition, chin joystick or a brain-computer interface (BCI) . The resulting MASSiVE control architecture with special emphasis on the HMI component is depicted in Fig. 2. Here, the deliberator has been moved to the sequencer component, and the HMI has direct access to control the actuators in the reactive layer during user interactions (e.g., to move the camera, until the desired object to be manipulated is in field of view).

Besides the focus on semi-autonomous system control, the MASSiVE framework includes a second main paradigm, namely the pre-structuring of task knowledge. This task planner input is specified offline in a scenario and model driven approach with the help of so-called process-structures on two levels of abstraction, the abstract level and the elementary level . After specification and before being used for task execution, the task knowledge is verified offline, to guarantee a robust runtime behavior. This development process model provides a structured guidance and enforce consistency throughout the whole process, so that uniform implementations and maintainability are achieved. Furthermore it guides through development and test of system core functionality (*skills*). The whole paradigm is depicted in Fig. 3.

The tasks is selected and started by the user via the HMI, depicted in Fig. 4, on a high level of abstraction, e.g. "cook meal". After initial monitoring to define the current state of the system and the environment, the tasks execution is performed and a list of elementary operations are created which can be executed autonomously by the system. These elementary operations consists of, e.g. image processing algorithm to recognize objects in the environment or manipulative algorithms to calculate a special trajectory to grasp an object.

Besides these layers, a world model is included in the control architecture that contains the current system's perspective on the world according to the task to be executed. Due to the hybrid architecture a separation of world-model data into two categories is mandatory: The deliberator operates with *symbolic* object representations (e.g. *"C"* for the representation of a cup), while the reactive layer deals with the sensor percepts taken from these objects, so-called *sub-symbolic* information. Examples for sub-symbolic information are the color, size, shape, location or weight of an object.

FRIEND::Image Processing - ROVIS

The main problem with service robotic systems such as the care-providing robot FRIEND is that they have to operate in dynamic surroundings where the state of the environment is unpredictable and changes stochastically, hence two main problems have been encountered when developing image processing systems for service robotics: unstructured environment and variable illumination conditions. They have to cope with a large amount of visual information and for the implementation of the vision system a high degree of complexity is necessary. A second major problem in robot vision is the wide spectrum of illumination conditions that appear during the on-line operation of the machine vision system, since colors are one important attribute in object recognition. The human visual system has the ability to compute color descriptors that stay constant even in variable illumination conditions, which is not the case for machine vision systems. A key requirement in this field is the reliable recognition of objects in the robot's camera image, extraction of object features from the images and, based on the extracted features, subsequent correct object localization in a complex 3D environment so that these information can be used for reliable object grasping and manipulation.

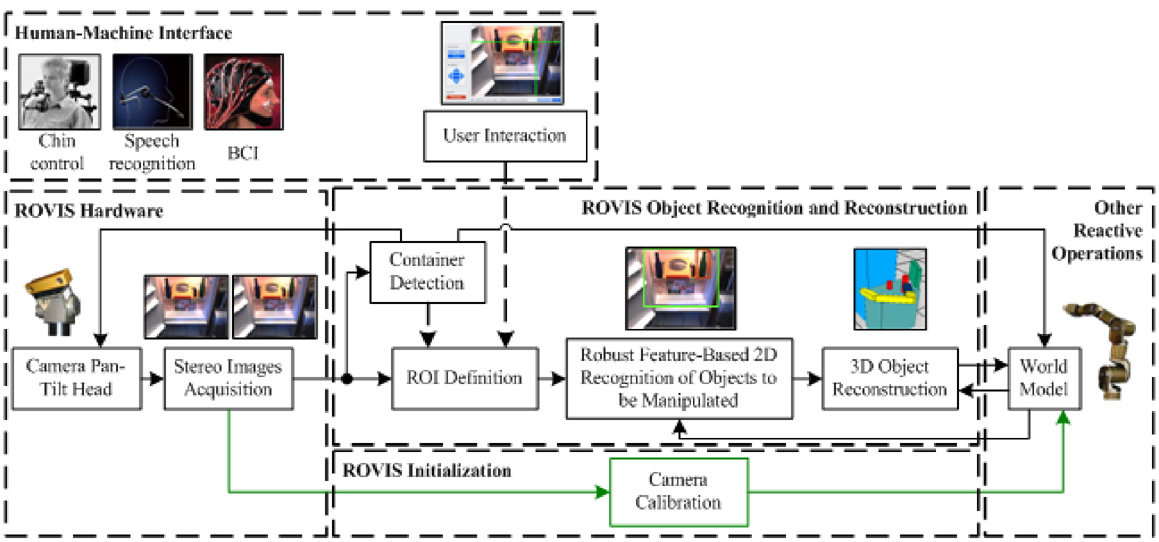


Fig. 5: Block diagram of ROVIS, the robust vision architecture for care-providing robot FRIEND.

In order to cope with the above described problems in the care-providing robot FRIEND the special vision system ROVIS (*RObust machine VIsion for Service robotics*) was developed. The structure of ROVIS is depicted in Fig. 5. There are two main ROVIS components: *hardware* and *object recognition and reconstruction chain*. The connection between ROVIS and the overall robot control system is represented by the World Model where ROVIS stores the processed visual information. At the initialization phase of ROVIS the extrinsic camera parameters needed for 3D object reconstruction and the transformation between stereo camera and manipulator which is necessary for vision based object manipulation are calculated by the *Camera Calibration* module. The object recognition and reconstruction chain consists of robust algorithms for object recognition and 3D reconstruction for the purpose of reliable manipulation motion planning and grasping in unstructured environments and variable illumination conditions. Therefore an accuracy of 5mm for the estimated 3D pose is necessary which enforces very good and precise algorithms. In ROVIS, robustness must be understood as the capability to the system to adapt to varying operational conditions and is realized through the inclusion feedback mechanisms at the image processing level and also between different hardware and software components of the vision system . A core part of the system is the automatic, closed-loop calculation of an image Region of Interest (ROI) on which vision methods are applied. By using a ROI the performance of object recognition and 3D reconstruction can be improved since the scene complexity is reduced.

Within ROVIS there are several methods used for object recognition, e.g. robust region based color segmentation and robust edge detection. The first one is for objects with uniform color and without texture (e.g. bottle, glass, handles) and the second one for objects with textures (e.g. books). In order to recognize objects in FRIEND's environment robustly special feature are extracted which are used to identify objects and to determine their pose . For big objects like refrigerator or microwave an improved SIFT (*Scale Invariant Feature Transform*) algorithm is used, which was developed at the Institute of Automation at the University of Bremen

FRIEND::Motion Planning

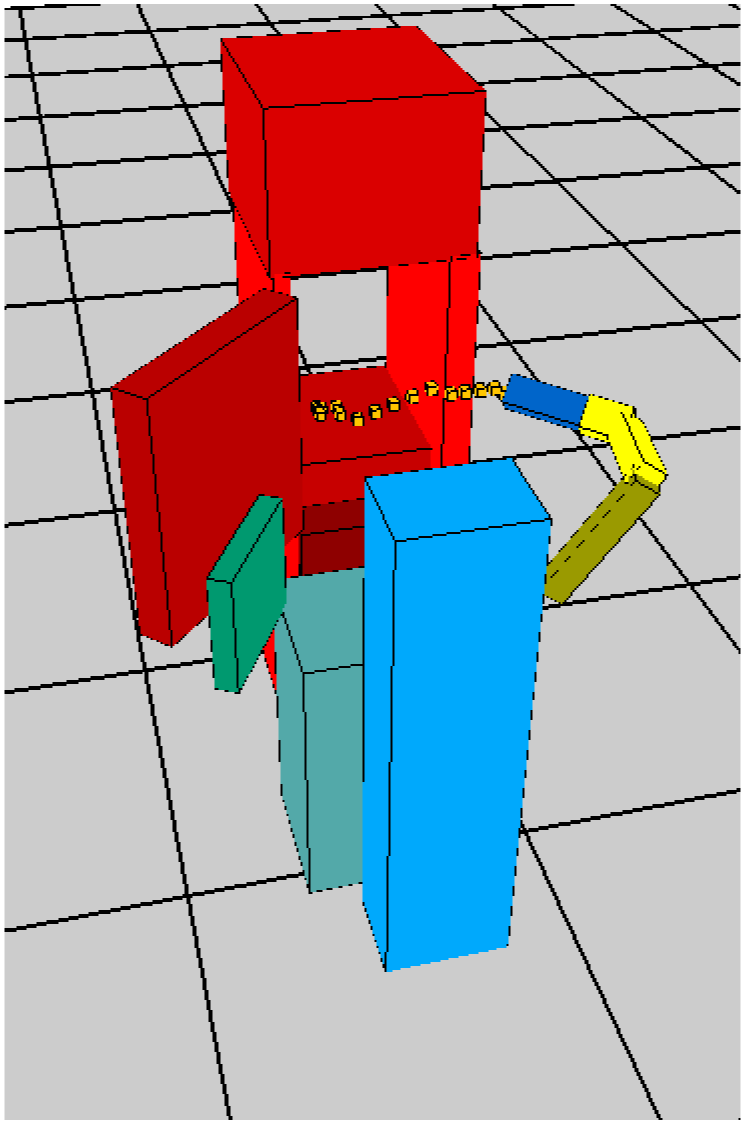


Fig. 6: A scene from the Mapped Virtual Reality (MVR) for a FRIEND task: move to the inside of a refrigerator.

Primary task of the motion planning is to find a collision free trajectory. Nevertheless, in service robotics, other aspects must be additionally taken into account like safety and smoothness of the trajectories. In order to suffice all these requirements for FRIEND a sensor-based motion planning for the manipulator was developed. The procedure is based on Cartesian space information , and therefore computationally efficient as well as with a high precision. The object grasping frames are calculated on-line which increases the flexibility of the system during execution. All implemented algorithms are suitable for real-time applications in service robotics but also for industrial usage.

The motion planning is performed based on a *Mapped Virtual Reality* (MVR), displayed in Fig. 6. This 3D-model of the environment is built from the information within the world model which was perceived by the sensors. Before any motion of the robot arm is performed the trajectory from the current to the goal configuration is calculated within this 3D-model. During motion on-line collision checking is done, since obstacles can move during motion. Obstacles are in this case all objects of the environment, which are not included in the current motion, also in some tasks the user in the wheelchair. It is important that there is at any time no danger for the user.

At the wrist of the manipulator a force-torque-sensor is installed. The information from this sensor is used to detect collisions or for fine-tuning during manipulative operations, e.g. when the robot arm should put a gripped object in a small opening. This ensures robustness during execution.

In general the trajectories calculated from the motion planning algorithms are robot like, i.e. clipped and jerky. In order to enhance this and make the movement of the manipulator more pleasant to the user the trajectories are smoothed and the quality is enhanced . Therefore the used robot arm is helpful. Since six degrees of freedom are sufficient to define a 3D pose in the environment (three degrees for position and three degrees for orientation) and FRIEND's manipulator has seven degrees of freedom, one degree of freedom can be chosen optional, i.e. the manipulator can turn its elbow joint by 360 degrees without changing gripper's pose. This can be used to find the best trajectory from a start to a goal configuration among an infeasible set of possible configurations. This seventh degree of freedom is also used to solve and avoid dead-locks during the motion process and to keep a minimal distance to obstacles.

Chapter- 5

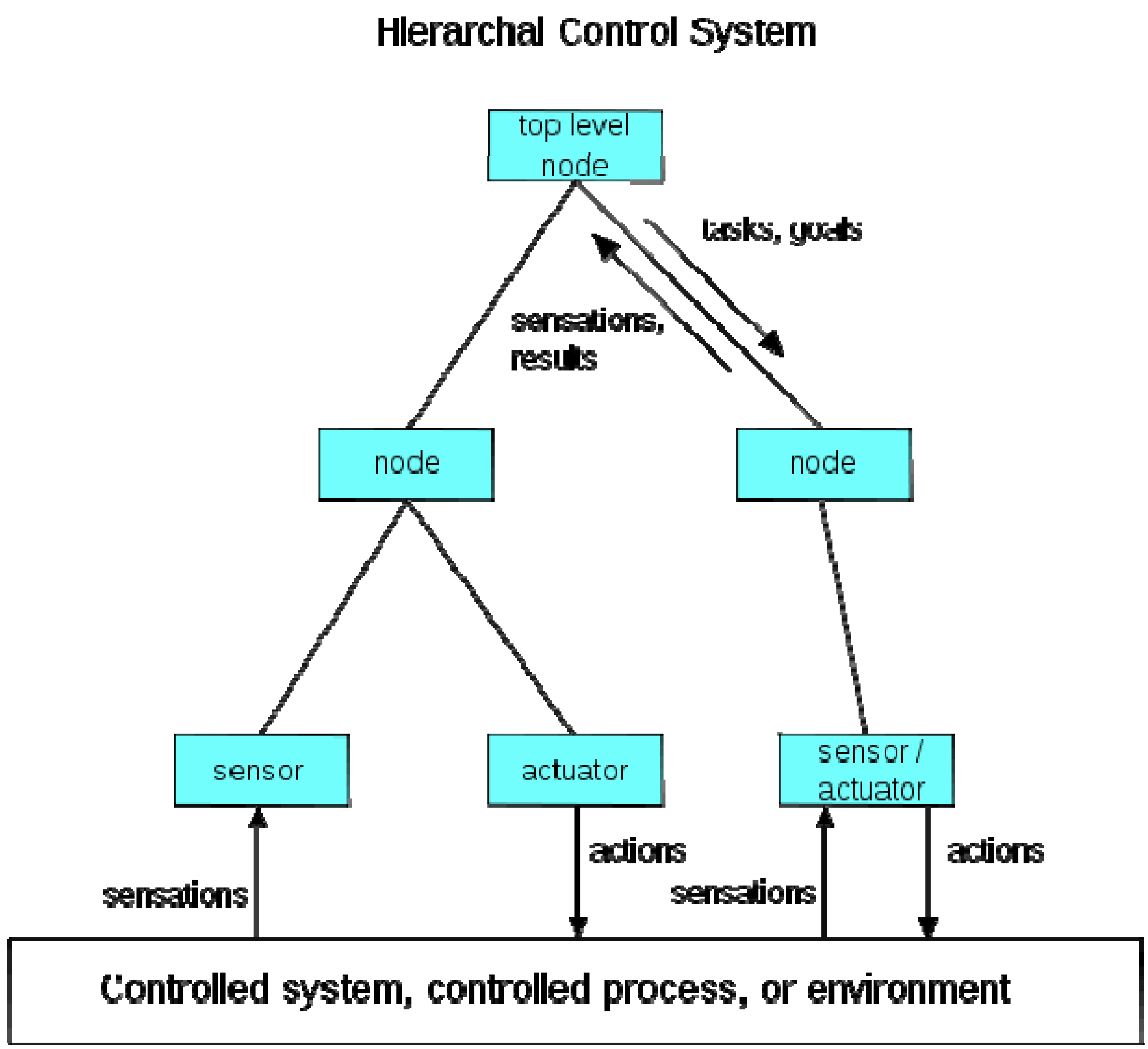
Important Robotics Art & Architecture

Distributed architecture for mobile navigation (DAMN)

The **Distributed architecture for mobile navigation (DAMN)** is a reactive robot architecture developed by Julio K. Rosenblatt at Carnegie Mellon University. DAMN consists of a collection of independently operating behaviors such as "go-to-goal" and "avoid obstacle", each holding a vote, weighted according to some mode manager. The votes are then counted by an arbiter, and the behavior with the most votes is chosen as the command to the vehicle controller. In DAMN, the decision-making process is distributed and asynchronous, without the need for an hierarchical structure.

Hierarchical control system

A **Hierarchical control system** is a form of Control System in which a set of devices and governing software is arranged in a hierarchical tree. When the links in the tree are implemented by a computer network, then that hierarchical control system is also a form of Networked control system.



A hierarchical control system takes the shape of a tree in which each node operates independently, performing tasks from its superior node, commanding tasks of its subordinate nodes, sending abstracted sensations to its superior node, and receiving sensations from its subordinate nodes. Leaf nodes are sensors or actuators.

Overview

A human-built system with complex behavior is often organized as a hierarchy. For example a command hierarchy has among its notable features the organizational chart of superiors, subordinates, and lines of organizational communication. Hierarchical control systems are organized similarly to divide the decision making responsibility.

Each element of the hierarchy is a linked node in the tree. Commands, tasks and goals to be achieved flow down the tree from superior nodes to subordinate nodes, whereas sensations and command results flow up the tree from subordinate to superior nodes.

Nodes may also exchange messages with their siblings. The two distinguishing features of a hierarchical control system are related to its layers.

- Each higher layer of the tree operates with a longer interval of planning and execution time than its immediately lower layer.
- The lower layers have local tasks, goals, and sensations, and their activities are planned and coordinated by higher layers which do not generally override their decisions. The layers form a hybrid intelligent system in which the lowest, reactive layers are sub-symbolic. The higher layers, having relaxed time constraints, are capable of reasoning from an abstract world model and performing planning. A hierarchical task network is a good fit for planning in a hierarchical control system.

Besides artificial systems, an animal's control systems are proposed to be organized as a hierarchy. In perceptual control theory, which postulates that an organism's behavior is a means of controlling its perceptions, the organism's control systems are suggested to be organized in a hierarchical pattern as their perceptions are constructed so.

Applications

Manufacturing, robotics and vehicles

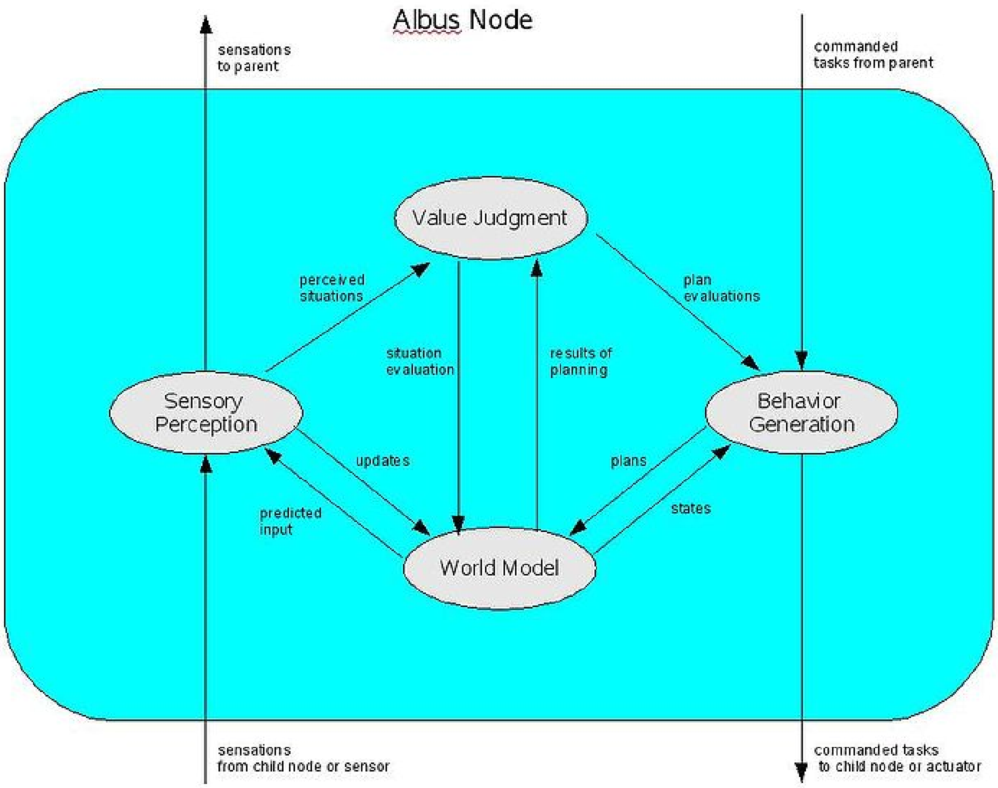
Among the robotic paradigms is the hierarchical paradigm in which a robot operates in a top-down fashion, heavy on planning, especially motion planning. Computer-aided production engineering has been a research focus at NIST since the 1980s. Its Automated Manufacturing Research Facility was used to develop a five layer production control model. In the early 1990s DARPA sponsored research to develop distributed (i.e. networked) intelligent control systems for applications such as military command and control systems. NIST built on earlier research to develop its Real-Time Control System (RCS) and Real-time Control System Software which is a generic hierarchical control system that has been used to operate a manufacturing cell, a robot crane, and an automated vehicle.

In November 2007, DARPA held the Urban Challenge. The winning entry, Tartan Racing employed a hierarchical control system, with layered mission planning, motion planning, behavior generation, perception, world modelling, and mechatronics.

Artificial intelligence

Subsumption architecture is a methodology for developing artificial intelligence that is heavily associated with behavior based robotics. This architecture is a way of decomposing complicated intelligent behavior into many "simple" behavior modules, which are in turn organized into layers. Each layer implements a particular goal of the software agent (i.e. system as a whole), and higher layers are increasingly more abstract. Each layer's goal subsumes that of the underlying layers, e.g. the decision to move forward by the eat-food layer takes into account the decision of the lowest obstacle-avoidance layer. Behavior need not be planned by a superior layer, rather behaviors may be triggered by sensory inputs and so are only active under circumstances where they might be appropriate. .

Reinforcement learning has been used to acquire behavior in a hierarchical control system in which each node can learn to improve its behavior with experience.



Constituents in a node from James Albus's Reference Model Architecture

James Albus, while at NIST, developed a theory for intelligent system design named the Reference Model Architecture (RMA) , which is a hierarchical control system inspired by RCS. Albus defines each node to contain these components.

- *Behavior generation* is responsible for executing tasks received from the superior, parent node. It also plans for, and issues tasks to, the subordinate nodes.
- *Sensory perception* is responsible for receiving sensations from the subordinate nodes, then grouping, filtering, and otherwise processing them into higher level abstractions that update the local state and which form sensations that are sent to the superior node.
- *Value judgment* is responsible for evaluating the updated situation and evaluating alternative plans.

- *World Model* is the local state that provides a model for the controlled system, controlled process, or environment at the abstraction level of the subordinate nodes.

At its lowest levels, the RMA can be implemented as a subsumption architecture, in which the world model is mapped directly to the controlled process or real world, avoiding the need for a mathematical abstraction, and in which time-constrained reactive planning can be implemented as a finite state machine. Higher levels of the RMA however, may have sophisticated mathematical world models and behavior implemented by automated planning and scheduling. Planning is required when certain behaviors cannot be triggered by current sensations, but rather by predicted or anticipated sensations, especially those that come about as result of the node's actions.

Unified Design Framework for Mobile Robot Systems

Overview

This design framework is concerned with the totality of the robot system, and is not constrained to components, architectures, middleware, hardware or software. It can be used to drive requirements analysis, design and implementation of new robot systems, or the classification of existing robot systems. Furthermore, it does not impose hardware, software or communication interfaces, therefore allowing flexible integration of new and legacy robot components. The framework is expressed in the Unified Modelling Language (UML), primarily using class diagrams.

Introduction

Mobile robot engineering encompasses techniques from a wide variety of scientific and engineering domains. Electronics, mechanics, computer science, biology, chemistry, physics and psychology have all played a significant role in what is now referred to as robotics. With such a diverse background it is difficult to define the term 'robot'. The problem is confounded by ambiguous descriptions and the overlapping of 'Robotics' with similarly unbounded scientific domains, such as 'Cybernetics' and 'Artificial Intelligence' . Intelligent, Autonomous and Unmanned are descriptions that immediately confuse not only the engineers of robot systems, but also the wider public. Mobile robots, such as wheeled robots and unmanned underwater vehicles, use a form of robot architecture. The term 'robot architecture' is commonly used to describe the software structure and its action selection methods . The robot architecture provides the robot command structure and has a wide-ranging effect on the robots ability to perform its desired tasks. Further to traditional control systems, a robot's architecture may be capable of performing deliberative actions. Traditionally, robot architectures are constrained to cognition and interaction with the vehicle. However, this work uses a unified design framework to describe the overall operation and structure of the robot system, which may include other robots, users and environments and various action selection techniques. Whereas traditionally, robot architectures have focused on abstraction of hardware and software elements, the proposed unified approach does not attempt to impose such boundaries.

Since the 1950s, three major paradigms for Robot Architectures have been proposed; these are commonly referred to as Deliberative, Reactive and Hybrid . Although these state-of–the-art techniques may meet the requirements imposed upon a robot, it is difficult for an engineer to identify the most appropriate techniques and design a robot system as a whole. This work is driven by three factors. Firstly, the robotics domain is flooded with robot architectures; it is desirable to focus the robotics community by identifying a framework, which accommodates these existing architectures. Secondly, providing a standard framework used in a variety of domains increases understanding, confidence and component re-use. This greater confidence can be transferred to systems which are otherwise difficult to test, such as spacecraft. Thirdly, designing robot systems is not an easy task. This paper provides engineering processes for requirements analysis, design, implementation and classification or robot systems. The design framework is expressed using the UML. This allows the robot system to be rigorously described in a visual form.

JAUS

Joint Architecture for Unmanned Systems. Formerly JAUGS (Joint Architecture for Unmanned Ground Systems), **JAUS** was originally an initiative by the United States Department of Defense to develop an open architecture for the domain of unmanned systems.

In order to ensure that the architecture is applicable to the entire domain of current and future unmanned systems, it is built on the five principles of vehicle platform independence, mission isolation, computer hardware independence, technology independence and operator use independence.

The JAUS Reference Architecture, which is no longer being maintained, is a component based message passing architecture that defines a data format and methods of communication between computing nodes. The architecture dictates a hierarchical system built up of subsystems, nodes and components, and contains a strictly defined message set to support interoperability. Significant portions of the architecture, including the definitions for subsystem, node and component, have been loosely defined in order to accommodate for the five principles on which it is based.

The architecture has migrated from the JAUS Working Group which was composed of individuals from the government, industry and academia to the Society of Automotive Engineers, Aerospace Division, Avionics Systems Division. The AS4, Unmanned Systems Technical Committee now maintains and advances the set of standards. The following standards have been migrated from the JAUS Reference Architecture to a services based framework:

AS5669, JAUS Transport Standard.

Defines packet construction addressing transport concerns including header compression, source/destination addressing, TCP, UDP and Serial links. AS5669 defines the format of a JAUS message as it flows between systems in an Ethernet (TCP and UDP) or serial data link.

AS5710, JAUS Core Service Set.

Establishes a common set of services for distributed systems communication and coordination. The Core Service Set includes service definitions for transport, events, access control, management, time, liveness and discovery.

AS6009, JAUS Mobility Service Set.

This standard migrates the primitive driver, waypoint and path segment drivers, along with the position/orientation components and messaging to the SAE JAUS set of standards.

Others currently in draft include:

- AS6040 Draft, HMI Services
- AS6057 Draft, Manipulation Services
- AS6062 Draft, Mission Services
- AS6060 Draft, Environment Sensing
- AS6063 Draft, USV Services

Another standard that evolved from the JAUS efforts is the “JAUS Service Interface Definition Language” or JSIDL. JSIDL standardizes the language for defining JAUS compliant interfaces. The specificati

Mibe architecture

MIBE architecture (*Motivated Independent BEhavior*) is a behavior-based robot architecture developed at Artificial Intelligence and Robotics Lab of Politecnico di Milano by Fabio La Daga and Andrea Bonarini in 1998. MIBE architecture is based on the idea of animat and derived from subsumption architecture, formerly developed by Rodney Brooks and colleagues at MIT in 1986.

Description

MIBE architecture is based on the belief that autonomy is grounded on motivation and autonomy should arise from superimposition of synergetic activities in response to multiple drives. An autonomous agent is developed to achieve several goals (primary goals), but secondary goals also originate as a consequence of environmental or functional constraints. In MIBE architecture both primary and secondary goals are handled in the same way and defined as *needs*. A specific drive originates from each need. MIBE architecture generates and weights all these drives in an explicit *motivational state*. The higher the urgency to satisfy a specific need, the higher its weight in the motivational state and the higher the drive to perform a behavior that satisfies the given need.

Differences between MIBE and subsumption architecture

MIBE architecture mainly departs from subsumption architecture due to the introduction of a top level *motivational structure* which determines behavior priorities at run time. That is, there are not layers and static hierarchical dependencies between behavioral modules, but each behavior constantly competes with others for taking control of the agent through the top level motivational state from which specific drives originate (via predetermined or reinforcement-learned functions).

While subsumption architecture is built on a predetermined hierarchy of behavioral modules, MIBE architecture consists of a more complex structure, where several behaviors (that always compete for taking control of the robot via the motivational state) can activate and control dynamically an adaptive set of underlying modules, called *abilities*. Each behavior performs its task by activating and tuning the abilities it needs. Abilities supply the functional modules for performing specific activities and may activate each other in a hierarchical structure in the same way behaviors use abilities. Both behaviors and abilities are implemented by the same kind of functional modules, but a fundamental difference exists: behaviors are *self-activating* modules in response to a robot+environment state, whilst abilities are just *functional blocks* activated and controlled by behaviors for accomplishing their tasks (or by higher-level abilities that have been already activated by a behavior). Behaviors exist for satisfying specific needs, whilst abilities are not related to any need, cause they are used by behaviors for accomplishing their tasks, but have no meaning alone.

The list of abilities needed by each module (behavior or ability) is represented by its *activation tree*; the complete set of activation trees can be represented by a system-wide acyclic *activation graph*.

Behaviors are activated on the basis of their specific drive pressure in the motivational state at run time: the most motivated behavior (i.e., the most urgent or convenient) is always activated. Nevertheless, less motivated behaviors could even be activated at the same time, with the constraint they cannot use any of the abilities already collected by a more motivated behavior which is already acting. The BCR subsystem (*Behavior Conflict Resolver*) ensures no conflicting behaviors (i.e., sharing one or more abilities in their activation trees) can be active at the same time.

The main advantage of MIBE architecture consists in its high *modularity*: new abilities and behaviors can be added easily without changing the existing modular structure. Similarly, a behavior can be modified or removed with no functional effects on other modules. Even *drives* can be added to/removed from the motivational structure or modified with no need to change the previous system structure, except rebalancing the drive-generation functions.

The main issue of MIBE architecture consists in defining the boundaries of the *state-space* by shaping the motivational structure (i.e.: tuning the drive-generation functions and/or their learning algorithms) so that the autonomous agent performs the right behavior for each robot+environment state.

Object Action Complex

Object-Action Complexes (OACs) are proposed as a universal representation enabling efficient planning and execution of purposeful action at all levels of a cognitive architecture (Kruger 2009, Worgotter 2008, Geib 2006, Piater 2009). OACs combine the representational and computational efficiency for purposes of search (the frame problem) of STRIPS rules and the object- and situation-oriented concept of affordance with the logical clarity of the event calculus. Affordance is the relation between a situation, usually including an object of a defined type, and the actions that it allows. While affordances have mostly been analyzed in their purely perceptual aspect, the OAC concept defines them more generally as state-transition functions suited to prediction. Such functions can be used for efficient forward-chaining planning, learning, and execution of actions represented simultaneously at multiple levels in an embodied agent architecture.

The PACO+ project, an Integrated Project funded by the European Commission through its Cognition Unit under the Information Society Technologies of the sixth Framework Programme (FP6), and launched on 1 February 2006, brings together an interdisciplinary research team to design and build cognitive robots capable of developing perceptual, behavioural and cognitive categories that can be used, communicated and shared with other humans and artificial agents. In the project they hypothesize that such understanding can only be attained by embodied agents and requires the simultaneous consideration of perception and action resting on three foundational assumptions:

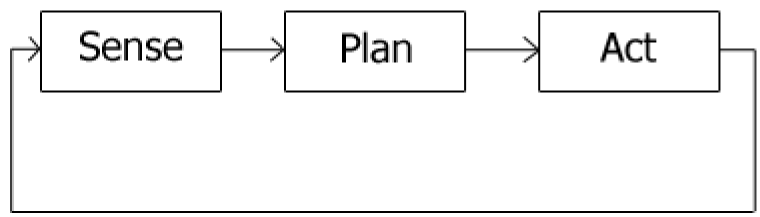
- Objects and Actions are inseparably intertwined in cognitive processing; that is "Object-Action Complexes" (OACs) are the building blocks of cognition.
- Cognition is based on reflective learning, contextualizing and then reinterpreting OACs to learn more abstract OACs, through a grounded sensing and action cycle.
- The core measure of effectiveness for all learned cognitive structures is: Do they increase situation reproducibility and/or reduce situational uncertainty in ways that allow the agent to achieve its goals?

Robotic paradigms

A **robotic paradigm** can be described by the relationship between the three primitives of robotics: Sense, Plan, and Act. It can also be described by how sensory data is processed and distributed through the system, and where decisions are made.

Hierarchical Paradigm

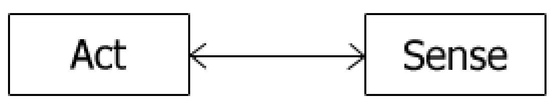
- The robot operates in a top-down fashion, heavy on planning.
- The robot senses the world, plans the next action, acts; at each step the robot explicitly plans the next move.
- All the sensing data tends to be gathered into one global world model.



Hierarchical Paradigm schema

Reactive Paradigm

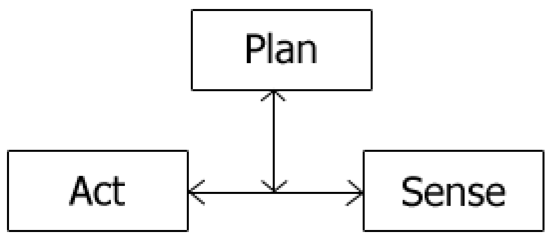
- Sense-act type of organization.
- The robot has multiple instances of Sense-Act couplings.
- These couplings are concurrent processes, called behaviours, which take the local sensing data and compute the best action to take independently of what the other processes are doing.
- The robot will do a combination of behaviours.



Reactive Paradigm schema

Hybrid Deliberate/Reactive Paradigm

- The robot first plans (deliberates) how to best decompose a task into subtasks (also called "mission planning") and then what are the suitable behaviours to accomplish each subtask.
- Then the behaviours starts executing as per the Reactive Paradigm.
- Sensing organization is also a mixture of Hierarchical and Reactive styles; sensor data gets routed to each behaviour the needs that sensor, but is also available to the planner for construction of a task-oriented global world model.



Hybrid Deliberate/Reactive Paradigm schema

Servo, subsumption, and symbolic architecture

The **Servo, subsumption, and symbolic (SSS) architecture** is a hybrid reactive/deliberative robot architecture developed by Jonathan H. Connell the IBM T.J. Watson Research Center.

Subsumption architecture

Subsumption architecture is a reactive robot architecture heavily associated with behavior-based robotics. The term was introduced by Rodney Brooks and colleagues in 1986 . Subsumption has been widely influential in autonomous robotics and elsewhere in real-time AI.

Description

A subsumption architecture is a way of decomposing complicated intelligent behaviour into many "simple" behaviour modules, which are in turn organized into layers. Each layer implements a particular goal of the agent, and higher layers are increasingly abstract. Each layer's goal subsumes that of the underlying layers, e.g. the decision to move forward by the eat-food layer takes into account the decision of the lowest obstacle-avoidance layer. As opposed to more traditional AI approaches subsumption architecture uses a bottom-up design.

For example, a robot's lowest layer could be "avoid an object", on top of it would be the layer "wander around", which in turn lies under "explore the world". Each of these horizontal layers access all of the sensor data and generate actions for the actuators — the main caveat is that separate tasks can suppress (or overrule) inputs or inhibit outputs. This way, the lowest layers can work like fast-adapting mechanisms (e.g. reflexes), while the higher layers work to achieve the overall goal. Feedback is given mainly through the environment.

Attributes of the architecture

The main advantages of the methodology are:

- the modularity,
- the emphasis on iterative development & testing of real-time systems in their target domain, and
- the emphasis on connecting limited, task-specific perception directly to the expressed actions that require it.

These innovations allowed the development of the first robots capable of animal-like speeds.

The main disadvantages of this model are:

- the inability to have many layers, since the goals begin interfering with each other,
- the difficulty of designing action selection through highly distributed system of inhibition and suppression, and
- the consequent rather low flexibility at runtime.

Three-layer architecture

The **Three-Layer Architecture** is a hybrid reactive/deliberative robot architecture developed by R. James Firby that consists of three layers: a reactive feedback control mechanism, a reactive plan execution mechanism, and a mechanism for performing time-consuming deliberative computations.