3D PRINTING

FIELDS OF STUDY

Computer Science; Digital Media

ABSTRACT

Additive manufacturing (AM), or 3D printing, comprises several automated processes for building three-dimensional objects from layers of plastic, paper, glass, or metal. AM creates strong, light 3D objects quickly and efficiently.

PRINCIPAL TERMS

- **binder jetting:** the use of a liquid binding agent to fuse layers of powder together.
- **directed energy deposition:** a process that deposits wire or powdered material onto an object and then melts it using a laser, electron beam, or plasma arc.
- **material extrusion:** a process in which heated filament is extruded through a nozzle and deposited in layers, usually around a removable support.
- **material jetting:** a process in which drops of liquid photopolymer are deposited through a printer head and heated to form a dry, stable solid.
- **powder bed fusion:** the use of a laser to heat layers of powdered material in a movable powder bed.
- **sheet lamination:** a process in which thin layered sheets of material are adhered or fused together and then extra material is removed with cutting implements or lasers.
- **vat photopolymerization:** a process in which a laser hardens layers of light-sensitive material in a vat.

Additive Manufacturing

3D printing, also called additive manufacturing (AM), builds three-dimensional objects by adding successive layers of material onto a platform. AM differs from traditional, or subtractive, manufacturing, also called machining. In machining, material is removed from a starting sample until the desired structure remains. Most AM processes use less raw material and are therefore less wasteful than machining.

The first AM process was developed in the 1980s, using liquid resin hardened by ultraviolet (UV) light. By the 2000s, several different AM processes had been developed. Most of these processes use liquid, powder, or extrusion techniques. Combined with complex computer modeling and robotics, AM could launch a new era in manufacture. Soon even complex mechanical objects could be created by AM.

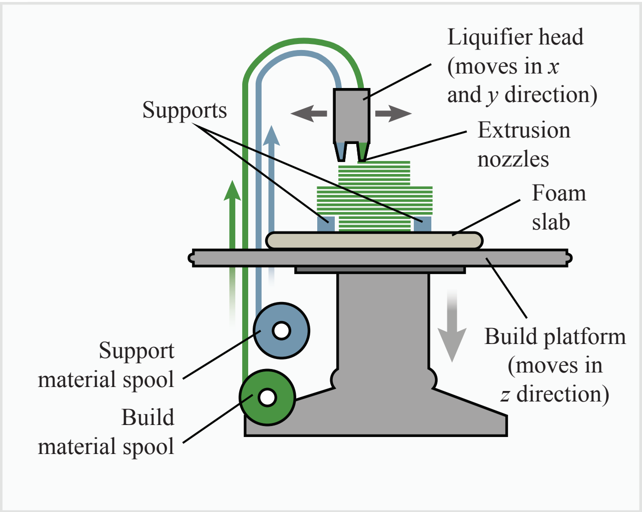
Software and Modeling

3D printing begins with a computer-aided design (CAD) drawing or 3D scan of an object. These drawings or scans are usually saved in a digital file format known as STL, originally short for "stereolithography" but since given other meanings, such as "surface tessellation language." STL files "tessellate" the object—that is, cover its surface in a repeated pattern of shapes. Though any shape can be used, STL files use a series of non-overlapping triangles to model the curves and angles of a 3D object. Errors in the file may need repair. "Slices" of the STL file determine the number and thickness of the layers of material needed.

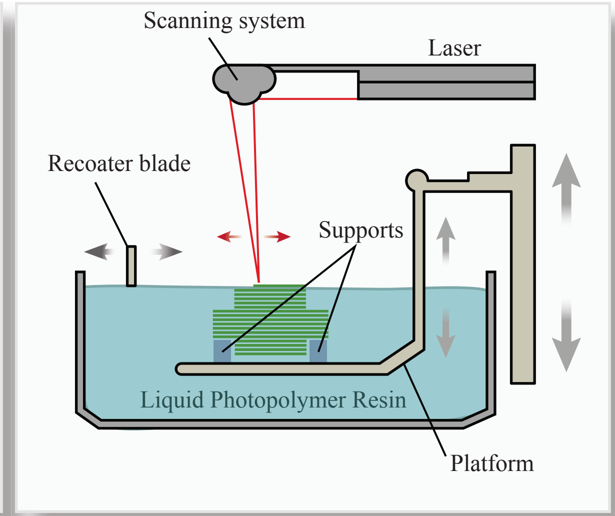
Liquid 3D Printing

The earliest AM technique was stereolithography (SLA), patented in 1986 by Chuck Hull. SLA uses liquid resin or polymer hardened by UV light to create a 3D object. A basic SLA printer consists of an elevator platform suspended in a tank filled with light-sensitive liquid polymer. A UV laser hardens a thin layer of resin. The platform is lowered, and the laser hardens the next layer, fusing it to the first. This process is repeated until the object is complete. The

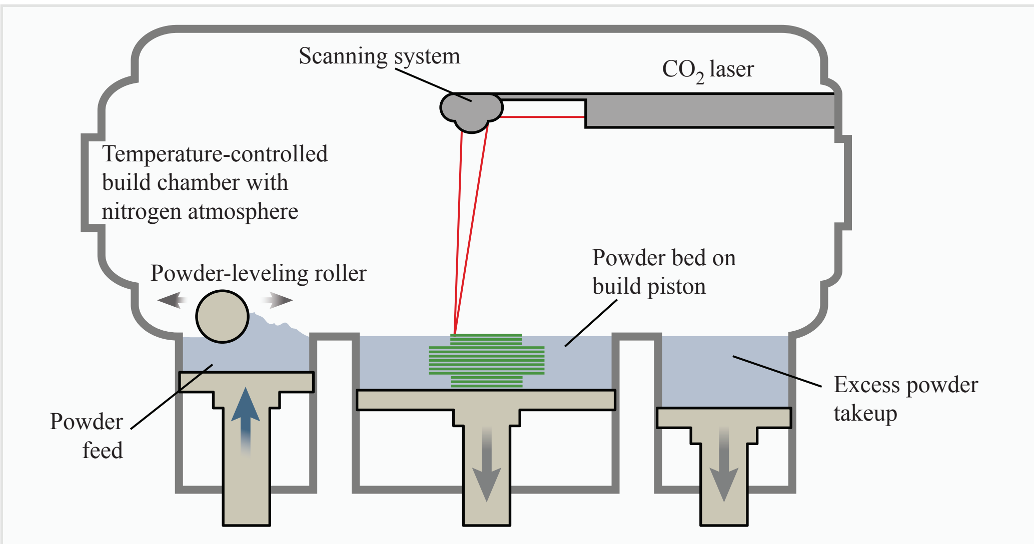
FDM Process



SLA Process



SLS Process



This presents a comparison of the three common 3-D printing processes: SLA (in which liquid polymer resin is solidified by a laser and support material is removed after completion), SLS (in which powder is fused by a CO2 laser and unfused powder acts as support), and FDM (in which liquid modeling material is extruded through extrusion nozzles and solidifies quickly, and a build material and a support material can be used in tandem, with the support material being removed after completion).

object is then cleaned and cured by UV. This AM technique is also called vat photopolymerization because it takes place within a vat of liquid resin. Various types of SLA printing processes have been given alternate names, such as “photofabrication” and “photo-solidification.”

Powder-Based 3D Printing

In the 1980s, engineers at the University of Texas created an alternate process that uses powdered solids instead of liquid. Selective layer sintering (SLS), or powder bed fusion, heats powdered glass, metal, ceramic, or plastic in a powder bed until the material is “sintered.” To sinter something is to cause its particles to fuse through heat or pressure without liquefying it. A laser is used to selectively sinter thin layers of the powder, with the unfused powder underneath giving structural support. The platform is lowered and the powder compacted as the laser passes over the object again.

Extrusion Printing

Material extrusion printing heats plastic or polymer filament and extrudes it through nozzles to deposit a layer of material on a platform. One example of this process is called fused deposition modeling (FDM). As the material cools, the platform is lowered and another layer is added atop the last layer. Creating extruded models often requires the use of a structural support to prevent the object from collapsing. Extrusion printing is the most affordable and commonly available 3D printing process.

Emerging and Alternative Methods

Several other 3D printing methods are also emerging. In material jetting, an inkjet printer head deposits liquefied plastic or other light-sensitive material onto a surface, which is then hardened with UV light. Another inkjet printing technique is binder jetting, which uses an inkjet printer head to deposit drops of glue-like liquid into a powdered medium. The liquid then soaks into and solidifies the medium. In directed energy deposition (DED), metal wire or powder is deposited in thin layers over a support before being melted with a laser or other heat source. Sheet lamination fuses together thin sheets of paper, metal, or plastic with adhesive. The resulting object is then cut with a laser or other cutting tool to refine the shape. This method is less costly but also less accurate than others.

The Future of 3D Printing

While AM techniques have been in use since the 1980s, engineers believe that the technology has not yet reached its full potential. Its primary use has been in rapid prototyping, in which a 3D printer is used to quickly create a 3D model that can be used to guide production. In many cases, 3D printing can create objects that are stronger, lighter, and more customizable than objects made through machining. Printed parts are already being used for planes, race cars, medical implants, and dental crowns, among other items. Because AM wastes far less material than subtractive manufacturing, it is of interest for conservation, waste management, and cost reduction. The technology could also democratize manufacturing, as small-scale 3D printers allow individuals and small businesses to create products that traditionally require industrial manufacturing facilities. However, intellectual property disputes could also occur more often as AM use becomes more widespread.

—Micah L. Issitt

Bibliography

“About Additive Manufacturing.” *Additive Manufacturing Research Group.* Loughborough U, 2015. Web. 6 Jan. 2016.

Hutchinson, Lee. “Home 3D Printers Take Us on a Maddening Journey into Another Dimension.” *Ars Technica.* Condé Nast, 27 Aug. 2013. Web. 6 Jan. 2016.

“Knowledge Base: Technologies in 3D Printing.” *DesignTech.* DesignTech Systems, n.d. Web. 6 Jan. 2016.

Matulka, Rebecca. “How 3D Printers Work.” *Energy.gov.* Dept. of Energy, 19 June 2014. Web. 6 Jan. 2016.

“The Printed World.” *Economist.* Economist Newspaper, 10 Feb. 2011. Web. 6 Jan. 2016.

“3D Printing Processes: The Free Beginner’s Guide.” *3D Printing Industry.* 3D Printing Industry, 2015. Web. 6 Jan. 2016.

A

ALGORITHMS

FIELDS OF STUDY

Computer Science; Operating Systems; Software Engineering

ABSTRACT

An algorithm is set of precise, computable instructions that, when executed in the correct order, will provide a solution to a certain problem. Algorithms are widely used in mathematics and engineering, and understanding the design of algorithms is fundamental to computer science.

PRINCIPAL TERMS

- **deterministic algorithm:** an algorithm that when given a input will always produce the same output.
- **distributed algorithm:** an algorithm designed to run across multiple processing centers and so, capable of directing a concentrated action between several computer systems.
- **DRAKON chart:** a flowchart used to model algorithms and programmed in the hybrid DRAKON computer language.
- **function:** instructions read by a computer's processor to execute specific events or operations.
- **recursive:** describes a method for problem solving that involves solving multiple smaller instances of the central problem.
- **state:** a technical term for all the stored information, and the configuration thereof, that a program or circuit can access at a given time.

An Ancient Idea

The term "algorithm" is derived from the name al-Khwarizmi. Muhammad ibn Musa al-Khwarizmi was a ninth-century Persian mathematician who is credited with introducing the decimal system to the West. He has been celebrated around the world as a pioneer of mathematics and conceptual problem solving.

"Algorithm" has no precise definition. Broadly, it refers to a finite set of instructions, arranged in a specific order and described in a specific language, for solving a problem. In other words, an algorithm is like a plan or a map that tells a person or a machine what steps to take to complete a given task.

Algorithm Basics

In computer science, an algorithm is a series of instructions that tells a computer to perform a certain function, such as sorting, calculating, or finding data. Each step in the instructions causes the computer to transition from one state to another until it reaches the desired end state.

Any procedure that takes a certain set of inputs (a data list, numbers, information, etc.) and reaches a desired goal (finding a specific datum, sorting the list, etc.) is an algorithm. However, not all algorithms are equal. Algorithms can be evaluated for "elegance," which measures the simplicity of the coding. An elegant algorithm is one that takes a minimum number of steps to complete. Algorithms can also be evaluated in terms of "goodness," which measures the speed with which an algorithm reaches completion.

Algorithms can be described in a number of ways. Flowcharts are often used to visualize and map the steps of an algorithm. The DRAKON computer language, developed in the 1980s, allows users to program algorithms into a computer by creating a flowchart that shows the steps of each algorithm. Such a flowchart is sometimes called a DRAKON chart.

Algorithms often specify conditional processes that occur only when a certain condition has been met. For instance, an algorithm about eating lunch might begin with the question, "Are you hungry?"

If the answer is “yes,” the algorithm will instruct the user to eat a sandwich. It will then ask again if the user is hungry. If the answer is still yes, the “eat a sandwich” instruction will be repeated. If the answer is “no,” the algorithm will instruct the user to stop eating sandwiches. In this example, the algorithm repeats the “eat a sandwich” step until the condition “not hungry” is reached, at which point the algorithm ends.

Types of Algorithms

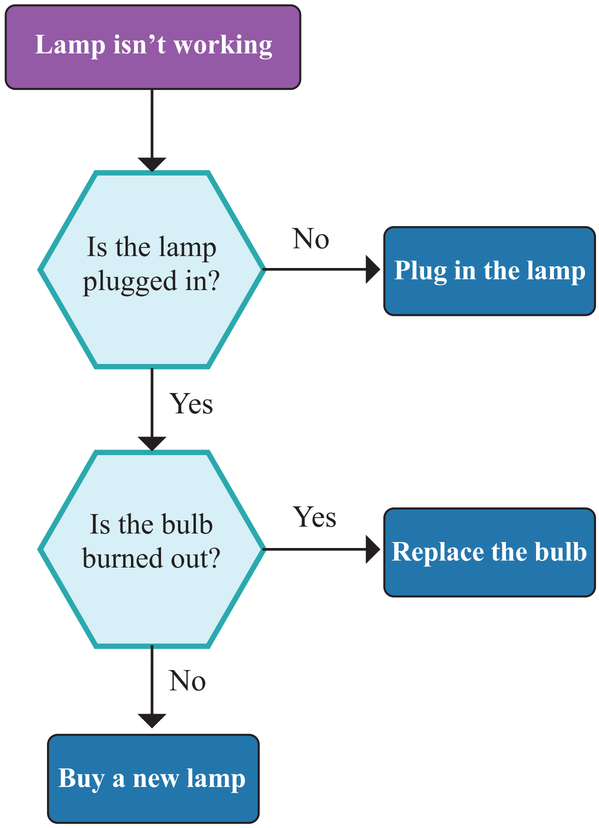
Various types of algorithms take different approaches to solving problems. An iterative algorithm is a simple form of algorithm that repeats the same exact steps in the same way until the desired result is obtained. A recursive algorithm attempts to solve a problem by completing smaller instances of the same problem. One example of a recursive algorithm is called a “divide and conquer” algorithm. This type of algorithm addresses a complex problem by solving less complex examples of the same problem and then combining the results to estimate the correct solution.

Algorithms can also be serialized, meaning that the algorithm tells the computer to execute one instruction at a time in a specific order. Other types of algorithms may specify that certain instructions should be executed simultaneously. Distributed algorithms are an example of this type. Different parts of the algorithm are executed in multiple computers or nodes at once and then combined.

An algorithm may have a single, predetermined output, or its output may vary based on factors other than the input. Deterministic algorithms use exact, specific calculations at every step to reach an answer to a problem. A computer running a deterministic algorithm will always proceed through the same sequence of states. Nondeterministic algorithms incorporate random data or “guessing” at some stage of the process. This allows such algorithms to be used as predictive or modeling tools, investigating problems for which specific data is lacking. In computational biology, for instance, evolutionary algorithms can be used to predict how populations will change over time, given estimations of population levels, breeding rates, and other environmental pressures.

Algorithm Applications

One of the most famous applications of algorithms is the creation of “search” programs used to find information on the Internet. The Google search engine can search through vast amounts of data and rank millions of search results in a specific order for different users. Sorting large lists of data was one of the earliest problems that computer scientists attempted to solve using algorithms. In the 1960s, the quicksort algorithm was the most successful sorting algorithm. Using a random element from the list as a “pivot,” quicksort tells the computer to pick other elements from the list and compare them to the pivot. If the element is less than the pivot, it is placed above it; if it is greater, it is placed below. The process is repeated until each pivot is in its proper place and the data is sorted into a list. Computer scientists are still attempting to find search and sorting algorithms that are more “elegant” or “good” in terms of completing the function quickly and with the least demand on resources.



An algorithm is a set of operations or a procedure for solving a problem or processing data. Flowcharts are often used as a visualization of the process, showing the order in which steps are performed.

Searching and sorting are the most famous examples of algorithms. However, these are just two of the thousands of algorithm applications that computer scientists have developed. The study of algorithm design has become a thriving subfield within computer science.

—*Micah L. Issitt*

BIBLIOGRAPHY

Anthes, Gary. "Back to Basics: Algorithms." *Computerworld*. Computerworld, 24 Mar. 2008. Web. 19 Jan. 2016.

Bell, Tim, et al. "Algorithms." *Computer Science Field Guide*. U of Canterbury, 3 Feb. 2015. Web. 19 Jan. 2016.

Cormen, Thomas H. *Algorithms Unlocked*. Cambridge: MIT P, 2013. Print.

Cormen, Thomas H., et al. *Introduction to Algorithms*. 3rd ed. Cambridge: MIT P, 2009. Print.

"Intro to Algorithms." *Khan Academy*. Khan Acad., 2015. Web. 19 Jan. 2016.

Toal, Ray. "Algorithms and Data Structures." *Ray Toal*. Loyola Marymount U, n.d. Web. 19 Jan. 2016.

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

FIELDS OF STUDY

Computer Science; Computer Engineering

ABSTRACT

The American Standard Code for Information Interchange (ASCII) is a character encoding system. It enables computers and other electronic communication devices to store, process, transmit, print, and display text and other graphic characters. Initially published in 1963, ASCII formed the basis for several other character encoding systems developed for use with PCs and the Internet.

PRINCIPAL TERMS

- **bit width:** the number of bits used by a computer or other device to store integer values or other data.
- **character:** a unit of information that represents a single letter, number, punctuation mark, blank space, or other symbol used in written language.
- **control characters:** units of information used to control the manner in which computers and other devices process text and other characters.
- **hamming distance:** a measurement of the difference between two characters or control characters that effects character processing, error detection, and error correction.
- **printable characters:** characters that can be written, printed, or displayed in a manner that can be read by a human.

UNDERSTANDING CHARACTER ENCODING

Written language, or text, is composed of a variety of graphic symbols called characters. In many languages, these characters include letters, numbers, punctuation marks, and blank spaces. Such characters are also called printable characters because they can be printed or otherwise displayed in a form that can be read by humans. Another type of character is a control character. Control characters effect the processing of other characters. For example, a control character might instruct a printer to print the next character on a new line. Character encoding is the process of converting characters into a format that can be used by an electronic device such as a computer or telegraph.

Originally designed for use with Samuel Morse's telegraph system, Morse code was one of the first character encoding schemes adopted for widespread use. Telegraphs transmit information by sending electronic pulses over telegraph wires. Morse code assigns each character to a unique combination of short and long pulses. For example, the letter *A* was assigned to the combination of one short followed by one long pulse, while the letter *T* was assigned to a single long pulse. Using Morse code, a telegraph operator can

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

This chart presents the decimal and hexidecimal ASCII codes for common characters on a keyboard.

send messages by transmitting a sequence of pulses. The sequence, or string, of pulses represents the characters that comprise the message text.

Other character encoding systems were created to meet the needs of new types of electronic devices including teleprinters and computers. By the early 1960s, the use of character encoding systems had become widespread. However, no standard character encoding system existed to ensure that systems from different manufacturers could communicate with each other. In fact, by 1963, over sixty different encoding systems were in use. Nine different systems were used by IBM alone. To address this issue, the American Standards Association (ASA) X3.4 Committee developed a standardized character encoding scheme called ASCII.

Understanding the ASCII Standard

The ASCII standard is based on English. It encodes 128 characters into integer values from 0 to 127. Thirty-three of the characters are control characters, and ninety-five are printable characters that include the upper- and lowercase letters from *A* to *Z*, the numbers zero to nine, punctuation marks, and a blank space. For example, the letter *A* is encoded as 65 and a comma as 44.

The encoded integers are then converted to bits, the smallest unit of data that can be stored by a computer system. A single bit can have a value of either zero or one. In order to store integers larger than one, additional bits must be used. The number of bits used to store a value is called the bit width. ASCII specifies a bit width of seven. For example, in ASCII, the integer value 65 is stored using seven bits, which can be represented as the bit string 1000001.

The ASCII seven-bit integer values for specific characters were not randomly assigned. Rather, the integer values of specific characters were selected to maximize the hamming distance between each value. Hamming distance is the number of bits set to different values when comparing two bit strings. For example, the bit strings 0000001 (decimal value 1) and 0000011 (decimal value 3) have a hamming distance of 1 as only the second to last bit differs between the two strings. The bit patterns 0000111 (decimal value 7) and 0000001 (decimal value 1) have a hamming distance of two as the bit in the third to last position also differs between the two strings. ASCII was designed to maximize hamming distance because larger hamming distances enable more efficient data processing as well as improved error detection and handling.

SAMPLE PROBLEM

ASCII defines the integer values for the first eleven lowercase letters of the alphabet as follows:

a = 97; *b* = 98; *c* = 99; *d* = 100; *e* = 101; *f* = 102; *g* = 103; *h* = 104; *i* = 105; *j* = 106; *k* = 107

Using this information, translate the word *hijack* to the correct ASCII integer values.

Answer:

The ASCII representation of the word hijack can be determined by comparing each character in the word to its defined decimal value as follows:

h	*i*	*j*	*a*	*c*	*k*
h (104)	*i* (105)	*j* (106)	*a* (97)	*c* (99)	*k* (107)
104	105	106	97	99	107

The correct ASCII encoding for the word hijack is 104 105 106 97 99 107.

Beyond ASCII

Following its introduction in 1963, ASCII continued to be refined. It was gradually adopted for use on a wide range of computer systems including the first IBM PC. Other manufacturers soon followed IBM's lead. The ASCII standard was also widely adopted for use on the Internet. However, as the need for more characters to support languages other than English grew, other standards were developed to meet this need. One such standard, Unicode can encode more than 120,000 characters. ASCII remains an important technology, however. Many systems still use ASCII. Character encoding systems such as Unicode incorporate ASCII to promote compatibility with existing systems.

—Maura Valentino, MSLIS

Bibliography

Amer. Standards Assn. *American Standard Code for Information Interchange*. Amer. Standards Assn., 17 June 1963. Digital file.

Anderson, Deborah. "Global Linguistic Diversity for the Internet." *Communications of the ACM* Jan. 2005: 27. PDF file.

Fischer, Eric. *The Evolution of Character Codes, 1874–1968*. N.p.: Fischer, n.d. *Trafficways.org*. Web. 22 Feb. 2016.

Jennings, Tom. "An Annotated History of Some Character Codes." *World Power Systems*. Tom Jennings, 29 Oct. 2004. Web. 16 Feb. 2016.

McConnell, Robert, James Haynes, and Richard Warren. "Understanding ASCII Codes." *NADCOMM*. NADCOMM, 14 May 2011. Web. 16 Feb. 2016.

Silver, H. Ward. "Digital Code Basics." *Qst* 98.8 (2014): 58–59. PDF file.

"Timeline of Computer History: 1963." *Computer History Museum*. Computer History Museum, 1 May 2015. Web. 23 Feb. 2016.

ANDROID OS

FIELDS OF STUDY

Computer Science; Operating Systems; Mobile Platforms

ABSTRACT

This article briefly discusses the general issues involved with mobile computing and presents a history and analysis of Google's Android operating system. It concludes with a look at Android's future in the growing market for mobile technology.

PRINCIPAL TERMS

- **application program interface (API):** the code that defines how two pieces of software interact, particularly a software application and the operating system on which it runs.
- **immersive mode:** a full-screen mode in which the status and navigation bars are hidden from view when not in use.
- **Material Design:** a comprehensive guide for visual, motion, and interaction design across Google platforms and devices.

- **multitasking:** in the mobile phone environment, allowing different apps to run concurrently, much like the ability to work in multiple open windows on a PC.
- **multitouch gestures:** touch-screen technology that allows for different gestures to trigger the behavior of installed software.

A FORCE IN MOBILE COMPUTING

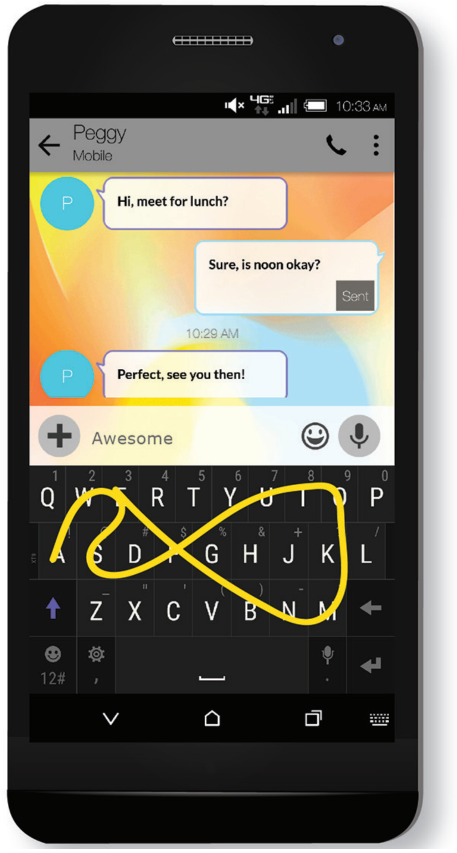
Mobile computing is the fastest-growing segment of the tech market. As pricing has become more affordable, developing nations, particularly in Africa, are the largest growing market for smartphones. With smartphones, users shop, gather information, connect via social media such as Twitter and Facebook, and communicate—one of the uses more traditionally associated with phones.

By far the most popular operation system running on mobile phones is Android. It has outpaced Apple's iOS with nearly double the sales. As of 2014, more than a million Android devices were being activated daily. Since its launch in 2008, Android has far and away overtaken the competition.

ANDROID TAKES OFF

Android came about amid a transformative moment in mobile technology. Prior to 2007, slide-out keyboards mimicked the typing experience of desktop PCs. In June of that year, Apple released its first iPhone, forever altering the landscape of mobile phones. Apple focused on multitouch gestures and touch-screen technology. Nearly concurrent with this, Google's Android released its first application program interface (API).

The original API of Google's new operating system (OS) first appeared in October 2008. The Android OS was first installed on the T-Mobile G1, also known as the HTC Dream. This prototype had a very small set of preinstalled apps, and as it had a slide-out QWERTY keyboard, there were no touch-screen capabilities. It did have native multitasking, which Apple's iOS did not yet have. Still, to compete with Apple, Google was forced to replace physical keyboards and access buttons with virtual onscreen controls. The next iteration of Android shipped with the HTC Magic and was accompanied by a virtual keyboard and a more robust app marketplace. Among the other early features that have stood the test of time are the pull-down notification list, home-screen widgets, and strong integration with Google's Gmail service.



The swype keyboard, originally designed for Android operating systems, was developed to speed up typing capabilities by allowing the user to slide a finger over the keyboard from letter to letter without lifting their finger to choose each character. This standard software in the Android operating system allows for quick texting.

One later feature, the full-screen immersive mode, has become quite popular as it reduces distractions. First released with Android 4.4, "KitKat," in 2013, it hides the navigation and status bars while certain apps are in use. It was retained for the release of Android 5.0, "Lollipop," in 2015.

ANDROID CHANGES AND GROWS

Both of Google's operating systems—Android and its cloud-based desktop OS, Chrome—are based on the free open-source OS Linux, created by engineer Linus Torvalds and first released in 1991. Open-source software is created using publicly available source code. The open-source development of

Android has allowed manufacturers to produce robust, affordable products that contribute to its widespread popularity in emerging and developing markets. This may be one reason why Android has had more than twice as many new users as its closest rival, Apple's iOS. This strategy has kept costs down and has also helped build Android's app marketplace, which offers more than one million native apps, many free of charge. By 2014 Android made up 54 percent of the global smartphone market.

This open-source development of Android has had one adverse effect: the phenomenon known as "forking," which occurs primarily in China. Forking is when a private company takes the OS and creates their own products apart from native Google services such as e-mail. Google seeks to prevent this loss of control (and revenue) by not supporting these companies or including their apps in its marketplace. Forked versions of Android made up nearly a quarter of the global market in early 2014.

Google's business model has always focused on a "rapid-iteration, web-style update cycle." By contrast, rivals such as Microsoft and Apple have had a far slower, more deliberate pace due to hardware issues. One benefit of Google's faster approach is the ability to address issues and problems in a timely manner. A drawback is the phenomenon known as "cloud rot." As the cloud-based OS grows older, servers that were once devoted to earlier versions are repurposed. Since changes to the OS initially came every few months, apps that worked a month prior would suddenly lose functionality or become completely unusable. Later Android updates have been released on a timescale of six months or more.

Android's Future

In 2014 alone, more than one billion devices using Android were activated. One of the biggest concerns about Android's future is the issue of forking. Making the code available to developers at no cost has made Android a desirable and cost-effective alternative to higher-end makers such as Microsoft and Apple, but it has also made Google a target of competitors.

Another consideration for Android's future is its inextricable link to the Chrome OS. Google plans to keep the two separate. Further, Google executives have made it clear that Chromebooks (laptops that run Chrome) and Android devices have distinct purposes. Android's focus has been on touch-screen technology, multitouch gesturing, and screen resolution, making it a purely mobile OS for phones, tablets, and more recently wearable devices and TVs. Meanwhile, Chrome has developed tools that are more useful in the PC and laptop environment, such as keyboard shortcuts. However, an effort to unify the appearance and functionality of Google's different platforms and devices called Material Design was introduced in 2014. Further, Google has ensured that Android apps can be executed on Chrome through Apps Runtime on Chrome (ARC). Such implementations suggest a slow merging of the Android and Chrome user experiences.

—Andrew Farrell, MLIS

Bibliography

Amadeo, Ron. "The History of Android." *Ars Technica*. Condé Nast, 15 June 2014. Web. 2 Jan. 2016.

"Android: A Visual History." *Verge*. Vox Media, 7 Dec. 2011. Web. 2 Jan. 2016.

Bajarin, Tim. "Google Is at a Major Crossroads with Android and Chrome OS." *PCMag*. Ziff Davis, 21 Dec. 2015. Web. 4 Jan. 2016.

Edwards, Jim. "Proof That Android Really Is for the Poor." *Business Insider*. Business Insider, 27 June 2014. Web. 4 Jan. 2016.

Goldsborough, Reid. "Android on the Rise." *Tech Directions* May 2014: 12. *Academic Search Complete*. Web. 2 Jan. 2016.

Manjoo, Farhad. "Planet Android's Shaky Orbit." *New York Times* 28 May 2015: B1. Print.

Newman, Jared. "Android Laptops: The $200 Price Is Right, but the OS May Not Be." *PCWorld*. IDG Consumer & SMB, 26 Apr. 2013. Web. 27 Jan. 2016.

Newman, Jared. "With Android Lollipop, Mobile Multitasking Takes a Great Leap Forward." *Fast Company*. Mansueto Ventures, 6 Nov. 2014. Web. 27 Jan. 2016.

APPLICATION

FIELDS OF STUDY

Applications; Software Engineering

ABSTRACT

In the field of information technology, an application is a piece of software created to perform a task, such as word processing, web browsing, or chess playing. Each application is designed to run on a particular platform, which is a type of system software that is installed on desktop computers, laptops, or mobile devices such as tablet computers or smartphones.

PRINCIPAL TERMS

- **app:** an abbreviation for "application," a program designed to perform a particular task on a computer or mobile device.
- **application suite:** a set of programs designed to work closely together, such as an office suite that includes a word processor, spreadsheet, presentation creator, and database application.
- **platform:** the specific hardware or software infrastructure that underlies a computer system; often refers to an operating system, such as Windows, Mac OS, or Linux.
- **system software:** the basic software that manages the computer's resources for use by hardware and other software.
- **utility program:** a type of system software that performs one or more routine functions, such as disk partitioning and maintenance, software installation and removal, or virus protection.
- **web application:** an application that is downloaded either wholly or in part from the Internet each time it is used.

Applications in Context

Applications are software programs that perform particular tasks, such as word processing or web browsing. They are designed to run on one or more specific platforms. The term "platform" can refer to any basic computer infrastructure, including the hardware itself and the operating system (OS) that manages it. An OS is a type of system software that manages a device's hardware and other software resources. Application designers may create different versions of an application to run on different platforms. A cross-platform application is one that can be run on more than one platform.

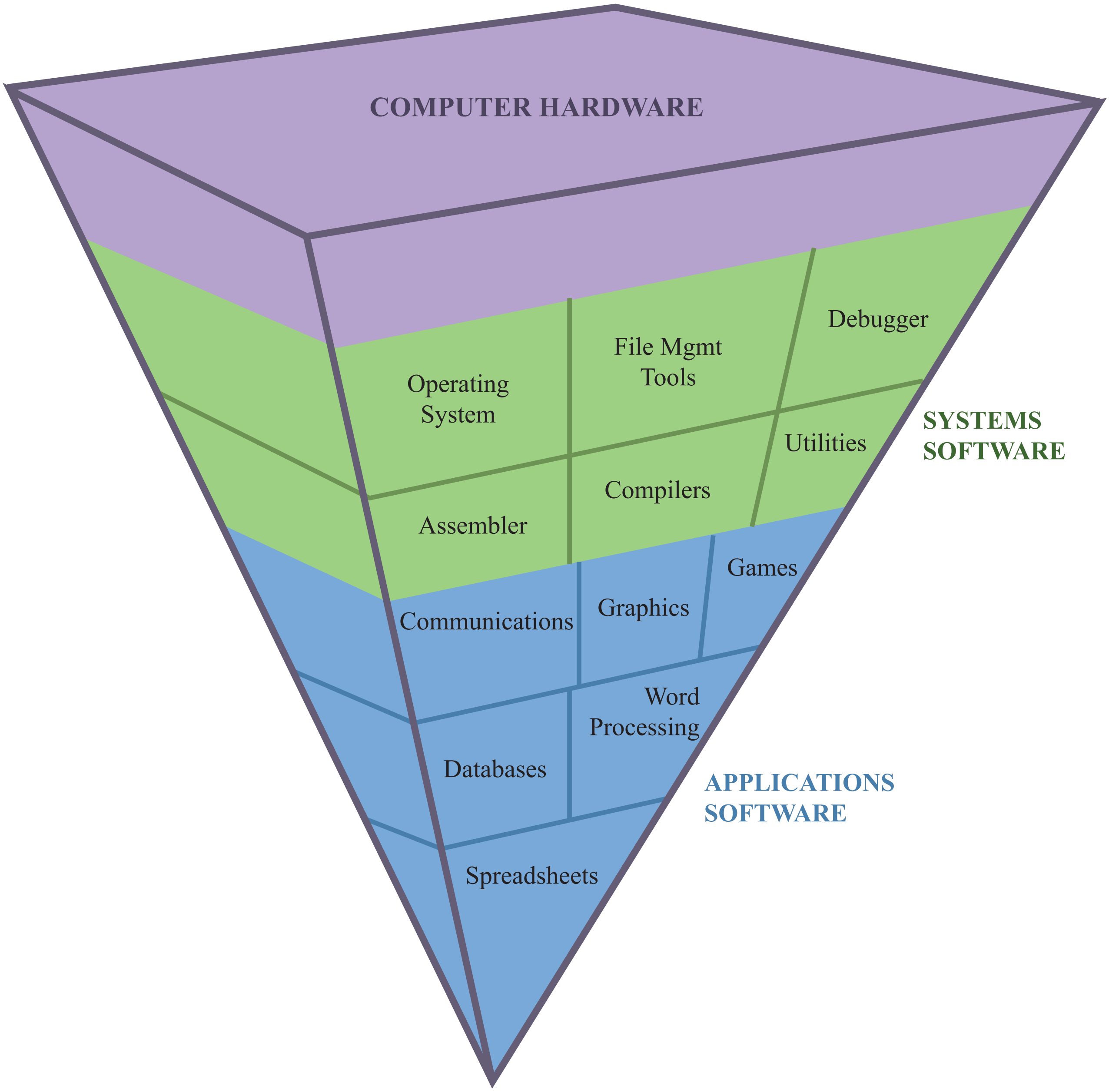
In the context of mobile devices such as tablets and smartphones, the term "application" is typically shortened to app. Since the introduction of the iPhone in 2007, apps have taken center stage in the realm of consumer electronics. Previously, consumers tended to be attracted more to a device's hardware or OS features. A consumer might have liked a certain phone for its solid design or fast processor, or they might have preferred the graphical interface of Microsoft Windows and Mac OS to the command-line interface of Linux. These features have since become much less of a concern for the average consumer. Instead, consumers tend to be more interested in finding a device that supports the specific apps they wish to use.

Evolution of Applications

Over the years, apps have become more and more specialized. Even basic utility programs that were once included with OSs are now available for purchase as separate apps. In some cases, these apps are a more advanced version of the utility software that comes with the OS. For example, an OS may come with free antivirus software, but a user may choose to purchase a different program that offers better protection.

Some software companies offer application suites of interoperating programs. Adobe Creative Cloud is a cloud-based graphic design suite that includes popular design and editing programs such as Photoshop and InDesign. Microsoft Office is an office suite consisting of a word processor (Word), a spreadsheet program (Excel), and other applications commonly used in office settings. These programs are expensive and can take up large amounts of storage space on a user's computer. Before broadband Internet access became widely available, application suites were distributed on portable media such as floppy disks, CD-ROMs, or DVDs, because downloading them over a dial-up connection would have taken too long.

As high-speed Internet access has become much more common, application developers have taken a different approach. Instead of investing in bulky application suites, users often have the option of using web applications. These applications run partly or



The variety and quantity of application software available is massive compared to the limited array of system software and hardware that support them.

entirely on remote servers, avoiding the need to install them on the computer's hard drive.

Types of Applications

Many different types of software fall under the broad heading of applications. A large segment of the application market is focused on office and productivity software. This category includes e-mail applications, word processors, spreadsheet software, presentation software, and database management systems. In an office environment, it is critical that users be able to create documents using these applications and share them with others. This often means that a business or organization will select a particular application suite and then require all employees to use it.

Other types of applications include games, audio-video editing and production software, and even software that helps programmers write new software. Due the complexity of software engineering, programmers have developed many applications to help them produce more polished, bug-free programs. Software developers may use multiple applications to code a single program. They might use a word processor or text editor to write the source code and explanatory comments, a debugging tool to check the code for errors, and a compiler to convert the code into machine language that a computer can execute. There is even a type of application that can emulate a virtual computer running inside another computer. These applications are often used by web-hosting companies. Instead of having to set up a new physical server for each customer that signs up, they can create another virtual server for the user to access.

Security Implications

Applications must have certain privileges in order to use the resources of the computer they are running on. As a result, they can sometimes be a point of weakness for attackers to exploit. A clever attacker can take over a vulnerable application and then use its privileges to make the computer behave in ways it should not. For example, the attacker could send spam e-mails, host illegally shared files, or even launch additional attacks against other computers on the same network.

Careers in Applications

Applications are the focus of a variety of career options for those interested in working with software. Apart from the obvious role of computer programmer, there are several other paths one might take. One option is quality assurance. Quality assurance staff are responsible for testing software under development to make sure it performs as it should. Technical support is another option. Technical support specialists assist users with operating the software and fixing errors it might cause. Yet another path is technical writing. Technical writers create software user manuals and training materials. Finally, some applications are so complex that using them can be a career in itself.

—Scott Zimmer, JD

Bibliography

Bell, Tom. *Programming: A Primer; Coding for Beginners.* London: Imperial Coll. P, 2016. Print.

Calude, Cristian S., ed. *The Human Face of Computing.* London: Imperial Coll. P, 2016. Print.

Dey, Pradip, and Manas Ghosh. *ComputerFundamentals and Programming in C.* 2nd ed. New Delhi: Oxford UP, 2013. Print.

Goriunova, Olga, ed. *Fun and Software: Exploring Pleasure, Paradox, and Pain in Computing.* New York: Bloomsbury, 2014. Print.

Neapolitan, Richard E. *Foundations of Algorithms.* 5th ed. Burlington: Jones, 2015. Print.

Talbot, James, and Justin McLean. *Learning Android Application Programming: A Hands-On Guide to Building Android Applications.* Upper Saddle River: Addison, 2014. Print.

AUTONOMIC COMPUTING

FIELDS OF STUDY

Computer Science; Embedded Systems; System-Level Programming

ABSTRACT

Autonomic computing is a subfield of computer science that focuses on enabling computers to operate independently of user input. First articulated by IBM in 2001, the concept has particular relevance to fields such as robotics, artificial intelligence (AI), and machine learning.

PRINCIPAL TERMS

- **autonomic components:** self-contained software or hardware modules with an embedded capacity for self-management, connected via input/outputs to other components in the system.
- **bootstrapping:** a self-starting process in a computer system, configured to automatically initiate other processes after the booting process has been initiated.
- **multi-agent system:** a system consisting of multiple separate agents, either software or hardware systems, that can cooperate and organize to solve problems.
- **resource distribution:** the locations of resources available to a computing system through various software or hardware components or networked computer systems.
- **self-star properties:** a list of component and system properties required for a computing system to be classified as an autonomic system.

Self-Managing Systems

Autonomic computing is a branch of computer science aimed at developing computers capable of some autonomous operation. An autonomic system is one that is, in one or more respects, self-managing. Such systems are sometimes described as “self-*” or “self-star.” The asterisk, or “star,” represents different properties of autonomic systems (self-organization, self-maintenance). Autonomic computing aims to develop systems that require less outside input, allowing users to focus on other activities.

Self-Star Systems

The concept of autonomic computing is based on autonomic systems found in nature. Examples of such systems include the autonomic nervous system of humans and the self-regulation of colonial insects such as bees and ants. In an autonomic system, the behaviors of individual components lead to higher-order self-maintenance properties of the group as a whole.

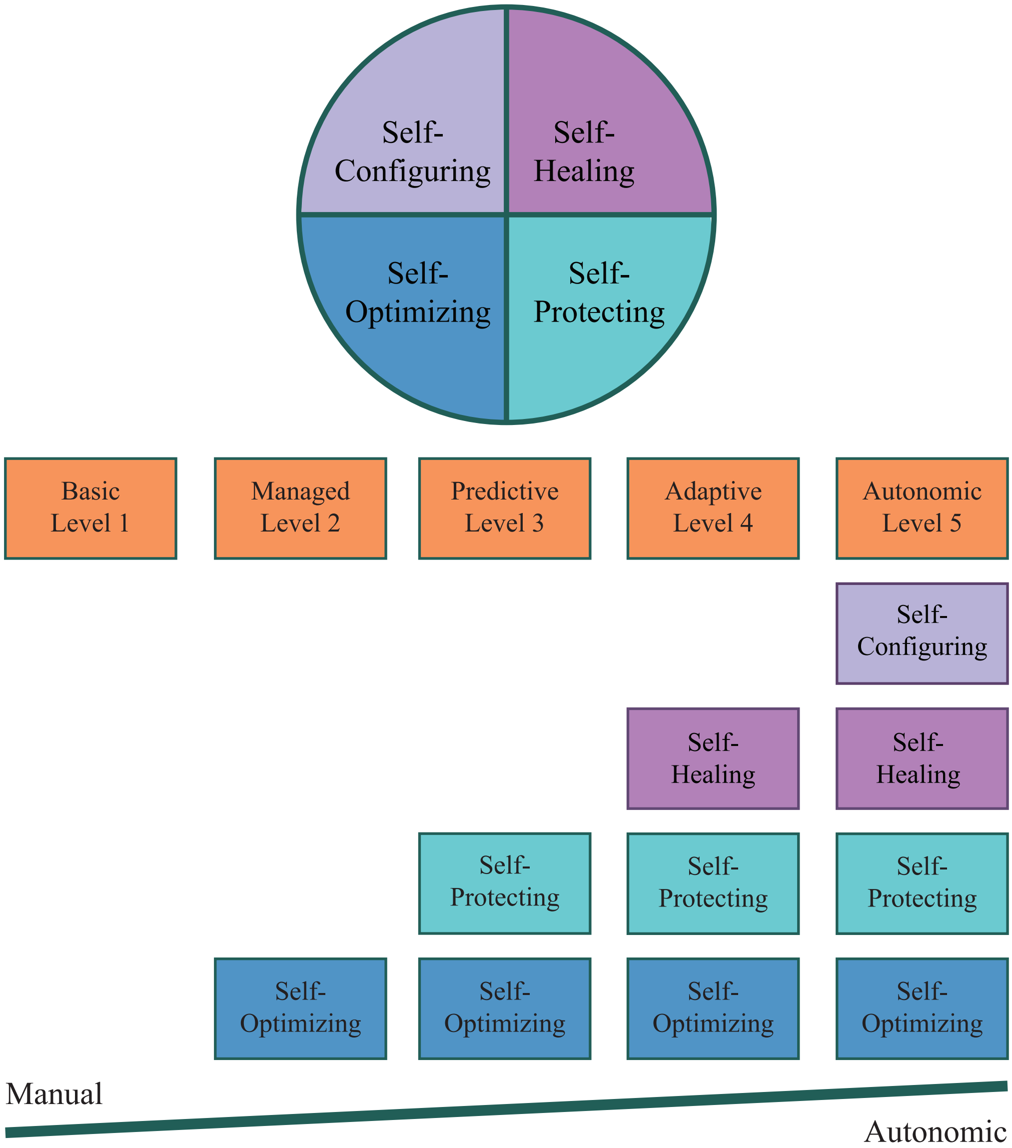
The properties that a system needs to function autonomically are often called self-star properties. One is self-management, meaning that the system can manage itself without outside input after an initial setup. Computer scientists disagree about what other self-star properties a system must have to be considered autonomic. Proposed properties include:

self-stabilization, the ability to return to a stable state after a change in configuration;
self-healing, the ability to recover from external damage or internal errors;
self-organization, the ability to organize component parts and processes toward a goal;
self-protection, the ability to combat external threats to operation; and
self-optimization, the ability to manage all resources and components to optimize operation.

Autonomic systems may also display self-awareness and self-learning. Self-awareness in a computer system differs from self-awareness in a biological system. In a computer system, self-awareness is better defined as the system’s knowledge of its internal components and configuration. Self-learning is the ability to learn from experiences without a user programming new information into the system.

Design of Autonomic Systems

An autonomic computer system is typically envisioned as having autonomic components (ACs), which are at least partly self-managing. An example of an AC is bootstrapping. Bootstrapping is the process by which a computer configures and initiates various processes during start-up. After a user turns on the computer, the bootstrapping process is self-managed. It proceeds through a self-diagnostic check and then activates various hardware and software components.



As computer systems have advanced from very basic technologies needing intense IT management toward autonomic systems that can self-manage, there have been four major stepping stones: self-optimizing, self-protecting, self-healing, and self-configuring. Each of these steps toward fully autonomic systems allows for more expansive computing while reducing the skill level required of the end users.

There are two basic models for autonomic computer design: a feedback control system and a multi-agent system. In a feedback control system, changing conditions provide feedback to the system that triggers changes in the system's function. Feedback control is often found in biological systems. In the autonomic nervous system, for example, levels of various neurotransmitters are linked to feedback systems that activate or deactivate ACs. A multi-agent system uses the collective functions of separate components to complete higher-order functions. For instance, groups of computers can be networked such that by performing individual functions, the components can collectively manage the system's functions and resources with reduced need for outside input. Linking multiple processors together changes the