

MUNK SÁNDOR
**SOFTWARE ROBOTS (SOFTBOTS), THEIR CHARACTERISTICS,
AND MILITARY APPLICATIONS**

"We are drowning in information but starved of knowledge."
John Naisbitt

Introduction

The global (and the military) information environment has already contains an enormous amount of information, and provides a lot of information services. These information, and services can, and has to be exploited to gain an information advantage (superiority), to achieve operational results. Computer technology has dramatically enhanced our ability to generate, store, and deliver information, but the tools for locating, filtering, and analyzing information have not kept pace. A popular solution for this situation is software robots. Software robots (softbots) are intelligent agents that use software tools and services on a person's behalf. Softbots allow a person to communicate what they want accomplished, and then dynamically determine how and where to satisfy the person's request. By acting as an intelligent personal assistant, softbots supports a qualitatively different kind of human-computer interface. A person can make a high level request, and the softbot uses search, inference, and knowledge to satisfy the request. Furthermore, the softbots are able to tolerate and recover from ambiguity, omissions, and errors in human requests.

1. Robots, and agents in general

Robots by definition (*The New Encyclopaedia Britannica, Volume 10 p.116.*) are automatically operated machines that replace human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner. The development of robotics began in the mid of the XX. century: the most relevant predecessors of robots were telemanipulators, servomechanisms, and numerical control (NC) machines. The first robot - Unimate - was built by George Devol in the beginning of the 1960s. Until that time there appeared different robot generations.

First-generation robots date from the 1960s and consisted of stationary, non-programmable, electromechanical devices without sensors. Second-generation robots were developed in the 1970s and contained sensors and programmable controllers. Third-generation robots appeared in the 1990s with sophisticated programming, speech recognition and/or synthesis, artificial intelligence properties, and other advanced features.

Robots can be grouped into different types by their mechanical, and software (self-control) complexity. Special types include nanorobots (with physical dimensions on the order of nanometers) for biological, chemical, or physical tasks; animats to model living animals; and androids, or cyborgs with their superficial resemblance to human beings.

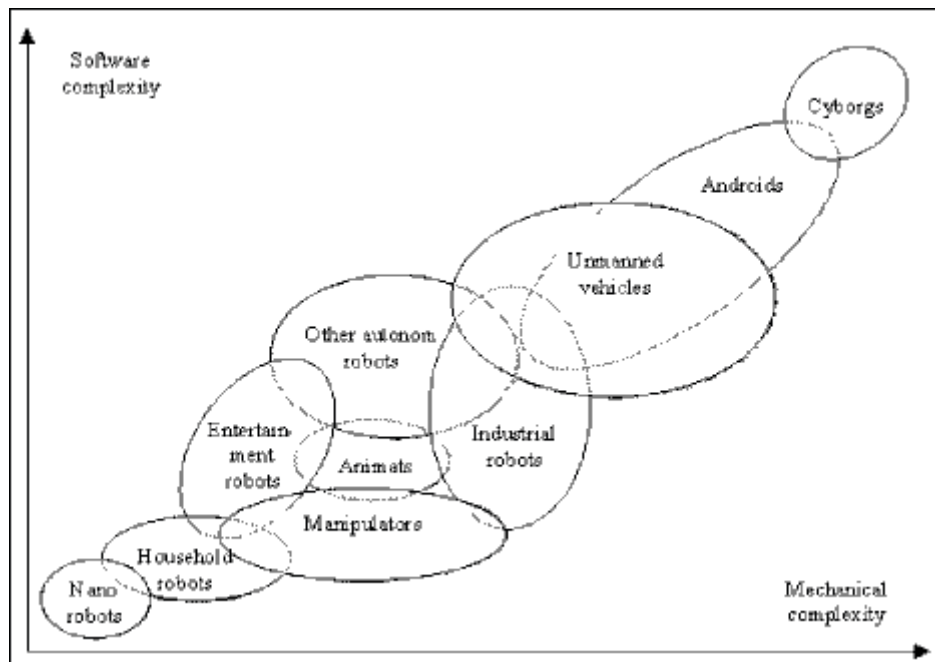


Figure 1 - Types of robots
(Source: Futó, 1999)

Robots were originally developed to do repetitive, "hard to do", or dangerous tasks in industry. In military applications the main reasons to use robots are the following (See *'Robotics, military' in International Military and Defense Encyclopedia*):

- reduced exposure of personnel to extreme danger in battle, or to other dangerous duties (such as rescue, decontamination, evacuation, and intelligence collection);
- personnel cost savings, either in operations or support, with a resulting reduction in life-cycle cost;
- reduced workload for skilled manpower categories (a single reduction of personnel requirements is not sufficient if robotics increase the required skill level ...);
- improved decision making under tactical stress conditions, including the introduction of artificial intelligence assistance, including collection, analysis and fusion, execution and elaboration of decisions, and dissemination.

At the end of the previous century the results got in different disciplines showed that robots as defined previously can be perceived as a subclass of a more abstract idea, the agents.

Nowadays there are a lot of similar, or synonymous terms used in this area, including: robots, bots, softbots (software robots), knowbots (knowledge robots, or knowledge-based robots), taskbots (task-based robots), userbots, personal agents, autonomous agents, intelligent agents, personal assistants, and many others. To analyze the similarities, and differences we can start with the everyday definition of agent - "one who acts for, or in place of, another". In this respect most (or all) robots are evidently agents, but they are not the only types of agents. The major classification of agents can be done by distinguishing animate organisms, real artifacts, and abstract constructs.

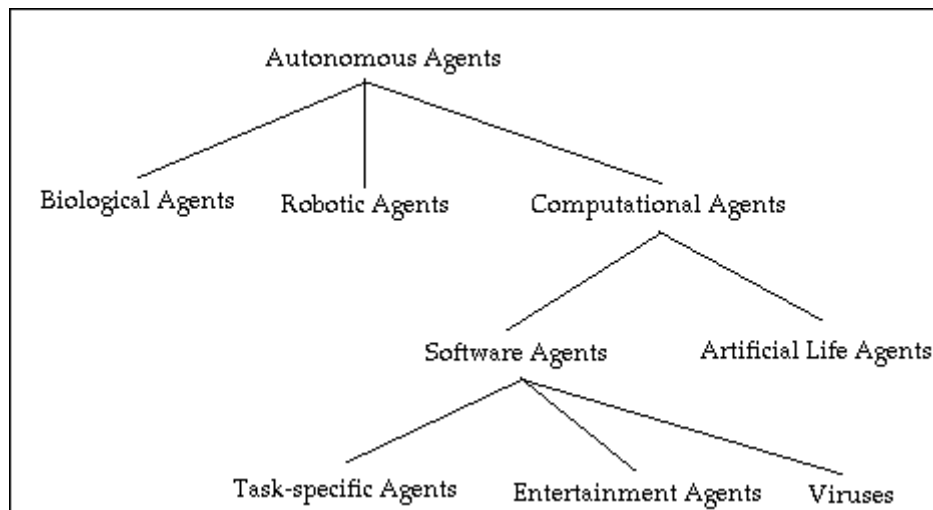


Figure 2 - Agent taxonomy
(Franklin and Graesser - 1996)

All the above agent types share some characteristics that constitute the essence of being an agent. Each agent is situated in, and is a part of some environment. Each senses its environment and acts autonomously upon it (no other entity is required to feed it input, or to interpret and use its output). Each acts in pursuit of its own agenda, whether satisfying evolved drives (as in humans and animals), or pursuing goals designed in by some other agent (as in software agents). Each acts so that its current actions may effect its later sensing, that is its actions effect its environment (it has feedback). Finally each acts continually over some period of time ("lives/works/runs until it decides not to"). The result is the following definition:

An autonomous agent is a system situated within and a part of an environment that senses that environment, and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future

(Franklin and Graesser - 1996)

So the basic characteristics of an agent are: situatedness, autonomy, reactivity, and temporal continuity. Reactivity means the ability to selectively sense and act upon. Autonomy means goal-directedness, proactive and self-starting behavior. Temporal continuity is a persistence of identity and state over long periods of time. There are also other characteristics that are useful for further classification of agents including the following: mobility, adaptivity, and cooperativity. All of these are descriptive characteristics that an object can possess at various degrees.

An influential white paper from IBM (Gilbert et al. - 1995) described agents in terms of a space defined by the three dimensions of **agency** (including autonomy), **intelligence** (including adaptivity), and **mobility**:

"**Agency** is the degree of autonomy and authority vested in the agent, and can be measured at least qualitatively by the nature of the interaction between the agent and other entities in the system. At a minimum, an agent must run asynchronously. The degree of agency is enhanced if an agent represents a user in some way ... A more advanced agent can interact with ... data, applications, ... services ... [or] other agents.

Intelligence is the degree of reasoning and learned behavior: the agent's ability to accept the user's statement of goals and carry out the task delegated to it. At a minimum, there can be some statement of preferences ... Higher levels of intelligence include a user model ... and reasoning ... Further out on the intelligence scale are systems that *learn* and *adapt* to their environment, both in terms of the user's objectives, and in terms of the resources available to the agent ...

Mobility is the degree to which agents themselves travel through the network ... Mo-bile scripts may be composed on one machine and shipped to another for execution ... [Mo-bile objects are] transported from machine to machine in the middle of execution, and carry-ing accumulated state data with them."

Agents can be further classified by degree of reactivity, and autonomy as follows: "A *reactive* agent reacts to changes in its environment or to messages from other agents ... An *intentional* agent is able to reason on its intentions and beliefs, to create plans of actions, and to execute those plans ... In addition to intentional agent capabilities, a *social* agent possesses explicit models of other agents." (Moulin-Chaib-draa - 1996)

2. Software robots (agents), and their characteristics

The original work on software agents was started to study computational models of distributed intelligence, but a new wave of interest has been fueled by the basic, and revolutionary changes in the information environment. This environment has already contains a continuously growing amount of information; provides more and more, and increasingly sophisticated information services; and has a complex technical infrastructure of computing, and communications technology devices, and networks. Neither the efficient use of information, and information services; nor the management of this infrastructure can be done without the help of new techniques, and methods, without using intelligent tools that assist in, or do autonomously information handling, processing, dissemination, and presentation tasks.

The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology. They had in view a system "that, when given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck. An agent would be a 'soft robot' living and doing its business within the computer's world." (Bradshaw - 1994)

A software agent is a program that carries out tasks for others, autonomously without being controlled by its master once the tasks have been delegated. The "others" may be human users, organizational processes, workflows, or applications (Farhoodi-Fingar - 1997). Software agents live in the information environment: they sense their environment by querying, monitoring other elements, or getting messages from them; and affect the environment, by sending messages, or requesting other active elements to do different tasks.

Although there is a tendency today to call every program (or at least every program with some degree of intelligence) agent, it is only a marketing strategy to improve sells. The situation is that every software agent per definitionem a program (or a program-complex), but not every

program is a (software) agent. A program to be an agent has to possess at least minimal "agency" characteristics, first of all a significant degree of autonomy.

Software agents have often called softbots (software robots), or shortly bots in information technology literature, and these terms are strongly connected with the Internet as the following definitions show:

A bot (short for "robot") is a program that operates as an agent for a user or another program or simulates a human activity. On the Internet, the most ubiquitous bots are the programs, also called spiders or crawlers that access Web sites and gather their content for search engine indexes. [WhatIs]

A bot is a program that autonomously explores the Internet for specific purposes. Derived from the word robot, a bot is a software program that performs repetitive functions, such as indexing information on the Internet. Robots that record and index all of the contents of a network to create searchable data-bases are sometimes called spiders or worms. ... A chatbot is another kind of bot that engages the user in conversations. [Agent Technology Glossary]

In the literature there are a lot of "bots" with different functions. The previous definitions contain and define two of them: spiders (crawlers), and chatbots. Two other interesting types are knowbots, and cobots.

A knowbot (knowledge-based robot) is a program that automatically searches Internet sites and gathers information according to user-specified criteria. A knowbot should not be confused with a search engine, crawler, or spider. A crawler or spider program visits Web sites and gathers information according to some generalized criteria and this information is then indexed so that it can be used for searching by many individual users.

A knowbot works with specific and easily changed criteria that conform to or anticipate the needs of the user or users. Its results are then organized for presentation but not necessarily for searching. An example would be a knowbot (sometimes also called a newsbot) that visited major news-oriented Web sites each morning and provided a digest of stories (or links to them) for a personalized news page (WhatIs.com).

A cobot (collaborative robot) is a robot designed to assist human beings as a guide or assistant in a specific task. A regular robot is designed to work more or less autonomously. In one approach to cobot design, the cobot allows a human to perform certain operations successfully if they fit within the scope of the task and to steer the human on a correct path when the human begins to stray from or exceed the scope of the task (WhatIs.com).

3. Military application of software agents

In general there are five different agent application areas of key interest to the military (The content of this point is mainly based on materials of a DARPA research project named Control of Agent-Based Systems - CoABS), where agents can:

- find, filter and present information for users;
- are aimed at complex problem-solving;
- manage, and operate different systems;
- provide services to other agents to help them cooperatively solve complex problems;
- and provide translational services between agents using different standards, communications protocols, languages, etc.

These areas are supported with different types of software agents. A functional classification of software agents contains: interface agents, information agents, task agents, system management agents, and infrastructure agents. The first three agent types closely related to human users' information handling and processing, the last two can be considered as "part of the system", so we will not discuss them in detail.

The overall purpose of software agents' application is to significantly cut the amount of time that warfighters spend manipulating information systems to allow them to have more time to focus on their mission. Different software agent types can be effectively used in almost every areas of command and control, and execution of military operations. Theoretically software agents can be used individually, but their most effective use is possible only in a collaborative environment.

Large-scale, cooperative teams, comprised of interacting agents from all groups, could offer new capabilities that are now beyond the realm of users, and software designers. An infrastructure that could provide these capabilities would allow software developers to design smaller pieces of code that would primarily function on solving problems via interaction with each other, rather than by each trying to duplicate functions provided by others.

In such a world, heterogeneous systems, separately developed, could be integrated into compound systems at run-time, based on the needs of the particular problems being solved. Finding these code pieces would be enabled by yellow page servers using taxonomies of common functionalities. Where gaps might exist between the agents, functions such as translation services could provide greater interoperability by seamlessly filling in the pieces. In addition, the entire "grid" of cooperating agents could be managed by brokering agents - which would help to manage the efficient flow of information across the grid. These agents could also provide tools for access control and information security, and they could provide a database allowing post hoc analysis of problem solving and other grid management services. Such a system would provide many capabilities that cannot be realized with any but the most state-of-the-art of today's tools. Users would be able to setup queries to search and filter large knowledge bases, to search through the net or other information sources, or to find computational resources needed for the problems they were trying to solve - all without needing to know the details of the underlying systems or information repositories. Current "legacy" systems could be brought to the grid through software wrappers and service descriptions., allowing their functionality to be tapped without major recoding. In addition, the cooperative nature of the problem solving, using existing software components, would allow both military and industrial users to develop large scale applications without large scale software development efforts.

Interface agents are software agents that provide assistance to a user dealing with a particular application. The metaphor is that of a personal assistant who is collaborating with the user in the same work environment (Maes - 1994). Interface agents operate within a user interface and actively assist the user in operating the interface and manipulating the underlying system. An interface agent is able to intercept the input from the user, examine it, and take appropriate action. While interface agents are not directly related to data management, they have the potential to play a large role in assisting users of data management systems. This becomes increasingly important as data management systems become more distributed and group together to form large, complex systems of systems. Agents in the interface can function as a bridge between domain knowledge about the data management systems and the user. These agents could assist users in forming queries, finding the location of data, explaining the semantics of the data among other tasks.

Interface agents are important components of an advanced military command and control

system. They seamlessly connect their users to the military information environment any-time, from anywhere: when a user actually enters the system, his/her interface agent informs the supporting environment where they are, what are they doing, how their resources are configured, which supplies they need, so the users immediately become part of a larger, coherent system. Interface agents autonomously gather information about the actual state, and services of the system, hiding the implementation details. They enable users to exploit information services without having to identify or supervise "step-by-step" procedures for accomplishing a given task.

Information agents are software agents that has access to at least one, and potentially many information sources, and is able to collate and manipulate information obtained from these sources to answer queries posed by users and other information agents (Wooldridge-Jennings - 1995). Information agents have different subtypes characterized with different functionality: search agents, filter-ing agents, mediator, and monitoring agents.

Information search agents help their users to effectively retrieve useful information, because the amount of information available stretches the capability of most users. To accomplish specific search tasks these agents contain domain knowledge about various information sources. This knowledge includes the types of information available at each source, how to access that information and other potentially useful knowledge such as the reliability and accuracy of the information source. Information search agents are basic components of an advanced military intelligence gathering system.

Information filtering agents attempt to deal with the problem of information overload by either limiting or sorting the information coming to a user. The basic idea is to develop an on-line surrogate for a user that has enough knowledge about the user's information needs so that it can select only those documents that would be of interest. These types of agents usually function as gatekeepers by preventing the user from being overwhelmed by a flood of incoming information. Filtering agents also work in conjunction with, or are sometimes incorporated into, search agents in order to keep the results from searches down to reasonable levels. Typically, filtering agents incorporate machine-learning mechanisms. This allows them to adapt to the needs of each user and to provide more precision than that typically provided by keyword filtering approaches.

Mediator agents provide the mechanisms that allow interoperate various data sources, and their users, provide access to heterogeneous data and knowledge bases. The data management landscape is populated with a multitude of different systems, most of which don't talk to each other. A mediator can be used to create a network of information gathering agents, each of which has access to one or more information sources. These agents use a higher level language, a communications protocol, and domain-specific ontologies for describing the data contained in their information sources. This allows each agent to communicate with the others at a higher semantic level. So mediator agents in military applications can wrap disparate data sources, or various message sources, originated from different components of a military force, or accessed from the military environment.

Monitoring agents constantly evaluate data sources and search for changes. In military applications they can evaluate any type of data: low level, e.g. sensory, data streams; formatted message flows; or processed situational data (from environmental even to everyday political events). The reason is that many tasks are dependent on the timely notification of changes in different data sources. A logistics planner may develop a plan for moving equipment from

one location to another, but the execution of that plan could be disrupted by the onset of bad weather at a refueling stop. The logistics planner would like to know of any events that would be likely to effect his plan as soon as they happen. Agents are useful for monitoring distributed data sources for specific data. Being software constructs, they have the patience necessary to constantly monitor data sources for changes. Alternately, mobile agents can be dispatched to remote or otherwise inaccessible locations to monitor data that the user might not normally have access to.

Task agents are software agents that support decision making executing information processing tasks. They have knowledge of the task domain, and which other task agents, or information agents are relevant to performing various parts of the task. In addition, tasks agents have strategies for resolving conflicts and fusing information retrieved by information agents. They perform most of the autonomous problem solving parts of information processing. A task agent (1) receives user delegated task specifications from an interface agent; (2) interprets the specifications and extracts problem solving goals; (3) forms plans to satisfy these goals; (4) identifies information seeking subgoals that are present in its plan; (5) decomposes the plans and coordinates with appropriate task agents or information agents for plan execution, monitoring, and results composition.

In military applications task agents can be used as teams of intelligent command and control agents with different skills. Teams will be constituted by different types of agents viewed as several subsets of homogeneous agents. Agents in different subsets have different skills. Agents will refine specified objectives, decompose the overall task according to their skills, organize themselves in order to enable collaboration, and learn to collaborate towards the most effective achievement of the team objectives.

The envisioned main integral part of these teams of intelligent agents consists of a pre-agreement on the task decomposition to organize the subteams of homogeneous agents and the collaboration during the autonomous task achievement. Agents will be equipped with techniques for run-time evaluation of the situation to decide between collaborating with other agents, or achieving the task individually. The agents must be able to work towards achieving concrete team objectives autonomously even in environments with limited, unreliable, or high-cost communication.

References:

Agent Technology Glossary. Revision 0.2 (1999) - Object Services and Consulting Inc.
[<http://www.objs.com/agility/tech-report/9903-agent-glossary.html>]

J. M. BRADSHAW (Ed.) (1997): Software Agents. - MIT Press.

DARPA Control of Agent Based Systems (CoABS) Program. CoABS Grid Vision Draft. (2000) - Global InfoTek Inc.

Faramarz FARHOODI - Peter FINGAR (1997): Competing for the Future with Intelligent Agents. In Distributed Object Computing Magazine.
[http://home1.gte.net/pfingar/agents_doc-rev4.htm]

Stan FRANKLIN - Art GRAESSER (1996): Is It an Agent or Just a Program? A Taxonomy for Autonomous Agents. - In Proceedings of the Third International Workshop on Agent

Theories, Architectures, and Languages. Springer Verlag.
Aparicio GILBERT et al.: The Role of Intelligent Agents in the Information Infrastructure. - IBM [<http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>]

Björn HERMANS (1997): Intelligent Software Agents on the Internet. - First Monday, Peer-Reviewed Journal on the Internet.

[http://www.firstmonday.dk/issues/issue2_3/ch_123/index.html]

International Military and Defense Encyclopedia. (1993) - Brassey's (US) Inc., Washington-New York.

Pattie MAES (1994): Agents that Reduce Workload and Information Overflow. In Communications of ACM, Vol 37 No 7.

B. MOULIN - B. CHAIB-DRAA (1996): Distributed Artificial Intelligence: an overview. In Foundations of Distributed Artificial Intelligence, Wiley.

The New Encyclopaedia Britannica. Micropaedia - Ready Reference. 15th Edition. (1991) - Encyclopaedia Britannica Inc., Chicago.

Whatis.com, The IT-Specific Encyclopedia. A techtarget.com Community. (2001)

[<http://www.techtarget.com>]

Michael WOOLDRIGDE - Nicolas R. JENNINGS (1995): Intelligent Agents: Theory and Practice. In Knowledge Engineering Review, Vol 10 No 2.