Software Agents

Stanislaw Stanek
Karol Adamiecki University of Economics in Katowice, Poland

Maciej Gawinecki
Systems Research Institute of the Polish Academy of Sciences, Poland

Malgorzata Pankowska
Karol Adamiecki University of Economics in Katowice, Poland

Shahram Rahimi
Southern Illinois University, USA

INTRODUCTION

The origins of the software agent concept are often traced back to the pioneers of artificial intelligence—John Mc Carthy, the creator of LISP programming language, and Carl Hewitt, the father of distributed artificial intelligence (DAI).

Kay (1984, p. 84) states that:

...the idea of an agent originated with **John McCarthy** *in the mid-1950s, 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.*

Nwana (1996, p. 205), on the other hand, claims that:

...software agents have evolved from multi-agent systems (MAS), which in turn form one of three broad areas which fall under DAI, the other two being Distributed Problem Solving (DPS) and Parallel Artificial Intelligence (PAI). (...) The concept of an agent (...) can be traced back to the early days of research into DAI in the 1970s – indeed, to **Carl Hewitt***'s concurrent Actor model. In this model, Hewitt proposed the concept of a self-contained, interactive and concurrently-executing object which he termed '***Actor***'. This object had some encapsulated internal state and could respond to messages from other similar objects*[1].

The software agent concept meant, in the first place, replacing the idea of an expert, which was at the core of earlier support systems, with the metaphor of an assistant. Until 1990s, decision support systems (DSS) were typically built around databases, models, expert systems, rules, simulators, and so forth. Although they could offer considerable support to the rational manager, whose decision making style would rely on quantitative terms, they had little to offer to managers who were guided by intuition. Software agents promised a new paradigm in which DSS designers would aim to augment the capabilities of individuals and organizations by deploying intelligent tools and autonomous assistants. The concept thus heralded a pivotal change in the way computer support is devised. For one thing, it called for a certain degree of intelligence on the part of the computerized tool; for another, it shifted emphasis from the delivery of expert advice toward providing support for the user's creativity (King, 1993).

Agents began focusing a good deal of attention in the mid-1990s. The outpour of publications on agents was triggered by Maes' (1994) article included in a special issue of the *Communications of the ACM* on "Agents that Reduce Work and Information Overload." Software agents were Maes's remedy for information overload and the increasing number of untrained users becoming forced to work with computers. Enormous enthusiasm aroused by the concept encouraged research focusing on know-how (formal modeling for software agents), organization (Foundation for Intelligent Physical

Table 1. Sample definitions of the software agent

Source	Definition
Maes (1994)	Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.
Russell & Norvig (1994)	An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A human agent has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors. A robotic agent substitutes cameras and infrared range finders for sensors and various motors for effectors. A software agent has encoded bit strings as its percepts and actions.
Ferber (1999)	An agent is a physical or virtual entity that is capable of acting in an environment, can communicate directly with other agents, is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimize), possesses resources of its own, is capable of perceiving its environment (but to a limited extent), has only a partial representation of this environment (and perhaps none at all), possesses skills and can offer services, may be able to reproduce itself, behavior tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representations, and the communications it receives.

Agents [FIPA]), standardization, instrumentation, and development of software agents (industry implementations examples). Results obtained, however, require thorough validation and further integration.

There is no commonly accepted definition of the term *software agent* (see Table 1). It is treated by some as a buzzword or an umbrella term comprising a number of varied, dynamically developing solutions, such as: collaborative, interface, mobile, information/Internet, reactive, hybrid, or smart agents (Nwana, 1996). Outside the realm of computers, agency means a relationship between two parties, one a principal and the other an agent who represents the principal in transactions with a third party. Other terms that are used as synonyms for software agent include: intelligent agent, software robot, knowbot (knowledge-based robot), softbot (intelligent software robot), taskbot (tasked-based robot), autonomous agent, and personal assistant.

BACKGROUND

An agent is a computational entity such as a software program or a robot that can be viewed as perceiving and acting upon its environment and that is autonomous, that is, its behavior at least partially depends on its own experience.

Some of the essential attributes that are emphasized in describing agent-based systems are:

- **Autonomy:** Autonomy means that, to an extent, agents can control and govern their own behavior and act without the intervention of humans or other systems.
- **Benevolence:** Benevolence means an agent always does what is asked of it by other agents or by humans.
- Intelligence: Intelligence indicates that agents can pursue their goals and execute tasks in a way that leads to maximizing its performance measures; intelligent agents can operate in a flexible and rational manner across a variety of contexts and environments.
- **Interactivity:** Interactivity implies that agents may be affected by other agents or by humans in pursuing their goals and executing their tasks.
- **Introspection:** Introspection implies that an agent is able to examine and reflect on its own thoughts, ideas, plans, goals.
- **Mobility:** Mobility indicates an agent can move either its code or its state across different environments, machines.
- **Pro-activity:** Pro-activity indicates an agent can not only exhibit goal-directed behavior but also take the initiative and get started on its own, stimulating itself to action.
- **Rationality:** Rationality indicates agents' actions are solely oriented on achieving a particular goal, therefore they will often behave in a way that is optimal for achieving that goal.
- **Reactiveness (reactivity):** Reactiveness indicates intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it.
- **Situatedness:** Situatedness implies an agent can continuously interact with, or be embedded in, its environment.