Intelligent Agents - Distinctive Features and Applications

Trayan Iliev

IPT – Intellectual Products & Technologies Ltd.
bl. 44, Slatinska Str., Sofia, Bulgaria
Tel.: (+359 2) 973 38 38
E-mail: tiliev@iproduct.org

Abstract: There are different definitions of the term "intelligent agent". Most of them emphasizing key properties such as autonomy, adaptability, communication and cooperation, reactivity, pro-activeness, and mobility. The intelligent agents can perceive the environment using sensors and act upon it using effectors in order to effectively achieve their goals. Depending on the presence of specialized hardware architecture can be distinguished hardware (robots) and software (softbots) agents. The Agent Communication Language (ACL) consists of three main components – knowledge representation format, ontologies for domain-specific vocabulary semantic definition, and a language for knowledge extraction and management KQML. It provides opportunities for implementation of complex and heterogeneous systems. The intelligent agents' applications include areas such as robotics, network resources management, mobile technologies, e-commerce and e-business, information access and filtering, e-learning, etc. Interesting new perspectives provides the research of "computer avatrs" – virtual human representatives in a digital world.

Keywords: agents, intelligent agents, adaptive agents, multiagent systems, types of agents, agent applications, ACL, information agents, interface agents, believable agents, personal assistants, emotional agents, avatars

People react to new technology in three different stages:
1. "It's crazy and don't waste my time"
2. "It's possible, but not worth pursuing"
3. "I've always said it was a good idea"

– Arthur C. Clarke

1. Introduction

I'm driving home after a working day. The garage door opens automatically because it recognizes my car using wireless communication channel. Entering the apartment I feel hungry so I go to the refrigerator. "Hi" – the refrigerator says – "how do you feel?". "O, I'm hungry, can you suggest something?" – I say. The refrigerator answers: "Yes I have a little surprise for you – while looking in Internet I have found a new receipt that perfectly suits to your taste. You may see how it looks in the kitchen computer screen, and it's balanced in nutrition ingredients. I've even purchased the needed products from the supermarket in order to surprise you. What do you think?". "Perfect! It looks great – how much time it will take to prepare it?" – am asking I. "O about 25 minutes – I'll tell the electric cooker the optimal temperature program. By the way the TV told me that it has recorded a nice concert play for you – would you like to watch it until the dish becomes ready?" – suggests the fridge. "No I have to read a paper and I will leave immediately after dinner. Please tell the TV to transfer the record in the car computer – I'll be listening there." – is my answer. The refrigerator says: "OK – arranged. Please put the plate into the cooker. I'll call you when the dish is ready. Wish you pleasant reading. And … the automatic vacuum cleaner complains it had too much work after the party last night, and asks to change the filter."

The situation described above may still look as a science fiction story, but the technologies that can make it reality already exist and have matured enough to be ready for practical application. The American company iRobot [70] (founded by the Director of the MIT Artificial Intelligence Lab Rod Brooks and his colleagues) already provides for the customer market an automatic vacuum cleaning machine called Roomba that "uses intelligent navigation technology to automatically clean nearly all household floor surfaces without human direction".

In order this dream of "the house of the future" to become a reality a number of models and technologies coming from different research areas should be combined – Systems Theory, Artificial Intelligence (speech recognition and synthesis, computer vision, distributed problem solving, planning, decision making and learning, etc.), Human-Computer Interaction (HCI), Object-Oriented Programming (OOP), Robotics, Communication Technologies (wireless communications, global networking), Social Sciences, Psychology. It is important that none of the technologies alone can provide the benefits coming from their synergetic application. The multidisciplinary nature of many contemporary research projects and the perceived need for more holistic approach towards the real-life, complex problems led to emergence of a new theoretical and practical approach for designing and building such systems, based on the concepts of *intelligent agents* and *multi-agent systems (MAS)*. This approach has tried to bring together the recent advances in each of the research areas mentioned above in order to facilitate their application for addressing problems intractable otherwise. It has been successfully applied in many domains, such as e-commerce and e-business, flexible networking, telecommunications and mobile technologies, adaptive human-computer interaction, information filtering, e-learning, entertainment, planning and scheduling, product design and manufacturing, traffic control, real-time control, etc.

The aim of this paper is to provide an introduction to the area of intelligent agents and MAS, by making comparisons with more traditional computational models, technologies, and software architectures.

2. Intelligent Agents and MAS – Historical Perspective

The first thing we can start with is that despite the numerous research publications and practical applications of intelligent agent technologies, there is no universally accepted definition of what we call "intelligent agent". It is not surprising that that such exact definition doesn't exist considering the number of paradigms covered by this dynamic and multidisciplinary approach. Before presenting some of the existing definitions of 'intelligent agent' and 'multi-agent system' concepts we will consider in more detail the main research areas that contributed to the emergence of this new paradigm.

2. 1. Artificial Intelligence (AI)

AI is probably the most important contributor to the intelligent agents paradigm. According to [1, p. 3], AI *"… attempts to understand intelligent entities. Thus, one reason to study is to learn more about ourselves. But unlike philosophy and psychology, which are also concerned with intelligence, AI strives to build intelligent entities as well as understand them"*. Essentially there are two complementary directions of research in the field of AI – building an intelligent (rational or human like) entities that can carry out tasks otherwise performed by humans, and using these artificial entities to better understand human intelligence. There are two main AI approaches developed in order to achieve these goals:

- symbolic approach – representing the domain objects, relations, actions, and goals symbolically and using the logic apparatus to manipulate them in symbolic form (production systems, frames, logical programming)
- behavioural (reactive) approach – based on the assumption that the intelligent system behaviour can emerge from interactions of a big number of simple, non-intelligent entities (artificial neural networks)

The first approach originates in the Newell and Simon's General Problem Solver (GPS) system [2], and expert systems are one of its most popular applications. By using it were developed algorithms for goal-directed searching in the space of possible decision alternatives, problem solving, and planning. A limitation of this approach is the so called property of "calculative rationality" [3, 4] in essence meaning that if the time for making optimal decision (plan) is not much shorter compared with the speed the environment changes, than the optimal decision generated for the initial state may not remain optimal for the current state of the environment when this decision has to be applied.

The second approach was developed in order to overcome the difficulties and limitations of the symbolic approach to cope with complex, real-time systems. Some problems not handled well by the symbolic approach are for example speech and image recognition, and analysis of real time data coming from multiple sensors. This approach was inspired by the observation of the mechanisms for achieving intelligent behaviour in biological organisms – the neurons have very simple structure but because of the number of connections between them the whole system may demonstrate intelligent behaviour. A natural limitation of this approach is that it is good in processing local data describing the current state of the environment, but there are some difficulties when trying to process data that is not local or current.

From historical perspective the AI research began by concentrating on several distinct problem areas concerned with the different aspects of the natural intelligence (making decisions, reasoning and planning, learning, speech recognition and synthesis, computer vision, etc.). The assumption was that achieving significant progress in each area it will be relatively easy to integrate such achievements. But there was recognized that complex, real-life problems require sufficient efforts for integrating different models and technologies in a complete system. What initially AI failed to recognize was the importance of the environment and the connections between different objects (active or passive) existing in it. For example most expert systems are working detached from the real physical environment they try to model, by using a human specialist as mediator. The recognition of these limitations led to emergence of *intelligent agents approach to AI*. Actually this new approach inherits all the achievements of traditional AI branches and supplies them with a useful overall metaphor pointing their natural place in modelling an entire intelligent entity – *intelligent agent*.

Though the characteristics of different intelligent agents are interesting by their own, even more exciting from the research point of view are the possibilities coming form modelling a community of such agents. Indeed the social behaviour can be considered one of the main factors in the evolution of human intelligence. Starting from the middle 1970s the study of *Distributed Artificial Intelligence (DAI)* has been rapidly evolving. It brings together (and is a driver for) achievements in many disciplines including AI, sociology, economy, philosophy and psychology. After the emergence "intelligent agents" paradigm the DAI field was transformed in similar way the AI field did, leading to research in *social agents* and *Multi-Agent Systems (MAS)*. A contemporary definition of DAI is given in [5, p. 1]: *"DAI is the study, construction, and application of multiagent systems, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks."*

As we can see from this definition the notion of "intelligent agents" and "multiagent systems" becomes central for DAI field too (in a similar way it consolidates the achievements in AI research).

2. 2. Object-Oriented Programming (OOP)

The object-oriented paradigm [6] has been another main source of influence in the formation of intelligent agents approach. The paradigm was initially proposed in the late 1960s, but it took more then 20 years for it to become a software development mainstream. Central for this paradigm is the notion of *object* and *class of objects*. The object is usually defined as having *internal state* defined by a number of *attribute values*, and well defined *behaviour* described in a form of set of *methods* or *messages* the object can handle and respond to. According to [7, p. 6]: *"A class is a group of objects that have something in common. A class captures a particular abstraction and provides a template for object creation."* – the classes can be view as moulds for creating objects of different types.

There are several important elements of the object model that distinguish it from other software engineering paradigms (procedure-oriented, logic-oriented, constraint-oriented, etc.). According to Booch [8] these are: *abstraction* – extraction of essential object's characteristics depending on the purpose of modelling; *encapsulation* – each objects provides well defined *interface specification* to other objects, hiding the implementation details (algorithms, data structures); *modularity – "the property of a system that has been decomposed into a set of cohesive and loosely coupled modules"*; *hierarchy* – there are two main hierarchies in the OOP approach – *class hierarchy* (describing the inheritance of the classes in the modelled problem area), and *object hierarchy* (describing the existing "whole-part" relationships between objects).

As defined in [8] there are also three minor elements: *typing* (the matching between parameter types actually send as arguments to a method with predefined argument types for that method), *persistence* (possibility to extend the existence of an object with given state either in time and space), and *concurrency*. In order to define *concurrency* Booch introduces the notion of *"active objects"* defined as follows: *"The object is a concept that unifies these two different viewpoints: each object (drawn from an abstraction of the real world) may represent a separate thread of control (a process abstraction). Such objects are called active. In a system based on an object-oriented design, we can conceptualize the world as consisting of a set of cooperative objects, some of which are active and thus serve as centres of independent activity."*. So the concurrency is a distinguishing property of active objects.

Black et al. in [9] state that *"an object model is appropriate for a distributed system because it implicitly defines (1) the units of distribution and movement and (2) the entities that communicate"*. As we will see these "active objects" and "entities that communicate" are very close to the concept of *mobile agents* described in the following sections.

3. Definitions

There are many definitions of what we call "intelligent agent" depending on the perspective and goals of different researchers. It is important to note that these definitions are not mutually exclusive, rather emphasizing the different aspects of the concept and its applications. Because of the great variety of definitions the authors often refer to particular kind of agents instead of speaking about intelligent agents in general. Comprehensive overviews of existing definitions can be found in [10] and [11].

In general the concept of "intelligent agent" can be defined by providing definitions of what we call "agent" and what we call "intelligence". Interestingly but the two concepts may be considered as quite orthogonal (independent of each other) – it is possible to have agents that are not very intelligent but are still useful (e.g. an agent that will automatically make software updates starting at given date and time simultaneously for all computers in a big, heterogeneous network), or programs that usually are not considered agents but are modelling certain aspects of the human intelligence (e. g. expert systems).

We may start the discussion about "agency" by looking at what people typicaly understand by using this term, and the types of human agents and agencies that have been existing. The Longman Dictionary [12] defines *"agent"* as *"a person who acts for or in the place of another by authority from him/her, a representative"*. According to this definition the central point for being an agent is the representation of somebody's interests and acting on his/her behalf. Murch and Johnson [11, p. 8] list more than 20 types of human agents (insurance agent, travelling agent, real estate agent, etc.) and state that central characteristics of such agents are their focus on the task, possession of special skills needed to accomplish it successfully, access to relevant information, existing contacts helping to provide a service, their effectiveness and efficiency providing the service (in less time and cost).

Another concept we need to address is *"intelligence"*. The discussions about what behaviours could be considered intelligent and about the nature of the *human and artificial intelligence* have been central for AI community for years [1, 13]. There is not commonly accepted definition of "artificial intelligence" (for definitions see [13, 14, 15, 16, 17, 18, and 19]) though it is recognized that processes and activities such as perceiving, acting, reasoning, making decisions, problem solving, and learning, play central role in intelligent behaviour realization.

In the rest of the section we will consider in more details some of the agent's definitions cited by Franklin and Graesser in [10].

3.1. AIMA agent

AIMA (Artificial Intelligence: A Modern Approach) agents are defined by Russel and Norvig in [1, p.31]: *"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors".*

This definition is too general – according to it an ordinary electric heater is an agent, because it can sense the environment through a sensor – switch on/off button, and act upon it by starting to heat. So the authors further define the *ideal rational agent* concept [1, p. 33]: *"For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has".*

According to Russel and Norvig each *rational agent* can be described by four characteristics: 1) *performance measure* – "defining the degree of success" of the agent, but external to the agent's reasoning mechanism; 2) *percept sequence* – perceptual history containing the information about the environment, sensed in the past; 3) built-in (or learned) *knowledge* about the environment; 4) *actions* the agent can perform in that environment.

They continue introducing the so called PAGE (Perceptions, Actions, Goals, Environment) characterization structure, allowing to compare agent applications automating tasks in different problem domains. For example the intelligent refrigerator agent described in the beginning of the paper would be characterized as follows:

- Perceptions: it should be able to detect the approaching people by using small camera, recognize their speech, and receive messages sent by other agents or servers (e.g. the TV agent, or a new cooking receipt from a Web site). It will also be equipped with multiple additional sensors enabling the automatic receiving of delivered cooking products from the supermarket, passing the meal to the user, etc. So the percepts will be pixels of varying intensity (and eventually colour), a sequence of discrete microphone signal values, textual messages coming from different agents/servers, and other electronic sensor signals.
- Actions: as described in the example dialog in the beginning of the paper the refrigerator agent should be able to: (1) generate speech output using speech generation engine, (2) visualize a meal picture and ingredients on a kitchen screen, (3) generate messages and requests (for products and information) to be send to other agents/servers using the local-area network or Internet, and (4)

control the mechanisms for receiving products, mixing them, and passing them to the user, by providing appropriate electrical signals.

- Goals: the important goal of the intelligent refrigerator agent will be to satisfy the needs of the user(s) by providing them with tasty and healthy food. Other goals may be defined as well: to provide information about the other agents work status, to provide a social comfort by communicating politely, and to entertain the user.
- Environment: as described above the environment will include users, other agents and Internet servers (e.g. the supermarket product delivery agent), as well as products and meals to be collected or produced.

The agent described above is quite sophisticated and it combines inputs and outputs from many sensors and effectors in a real world environment in order to be able to satisfy the formulated goals, that's why it may look slightly unrealistic. The contemporary intelligent agent applications and products usually concentrate on more limited set of features providing real value for the user. However in the future these features can be integrated in more advanced application architectures. Murch and Johnson [11, p. 39] anticipate the development of several agent generations – starting from host-based, standalone software applications, through communities of cooperating agents (e.g. in the e-commerce domain) towards mobile, negotiating, hierarchically structured agents, and real-world robots.

3.2. Maes agent

Pattie Maes from MIT Media Laboratory proposes the following definition of "autonomous agents" [20]: *"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."*. It emphasizes the *autonomy* as important agent characteristic. In addition to the ability to sense the environment and to act upon it *autonomous agents* should realize a *goal directed behaviour* in an environment that is *complex and dynamic*. There is not explicit requirement for agents to be able to perform high-level information processing activities such as planning, decision making or learning.

3.3. Hayes-Roth agent

According to Barbara Hayes-Roth from Stanford Knowledge Systems Laboratory [21]: "*Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions"*. This definition puts the explicit requirement for agents to be able to reason and solve problems in order to be called intelligent. Further the author stresses the importance of agent architecture defined as: *"the abstract design of a class of agents: the set of structural components in which perception, reasoning, and action occur, the specific functionality and interface of each component, and the interconnection topology among components"* in order the agents to be able to function (or exhibit appropriate *behaviour*) in certain *niches* or *"classes of operating environments"*. The recognition of the requirement that the three aspects of an agent-based system: structural (architecture), functional (behaviour) and contextual (niche, or class of the environment that agents occupy) need to be coordinated in order the system to function effectively, seems important not only addressing the intelligent agents development, but also for other engineering disciplines.

3.4. SodaBot agent

Michael Coen from MIT Artificial Intelligence Lab sets following minimum criteria for a program to be called *"software agent"* [71]: *1)Software agents engage in dialogs; 2) Software agents are* *autonomous and intelligent; 3) Software agents must be robust; 4) Software agents are generally not time invariant; 5)Software agents are typically distributed across a network, so their behaviour can have both local and global effect.* The approach of describing agents' characteristics instead of giving a strict definition of "agency" seems more appropriate and practically oriented given the variety of existing "agent-based" applications. We will try do outline the intelligent agents' characteristics identified by different researchers in the following sections.

3.5. Foner agent

Lenny Foner from MIT Media Lab states that *"a software agent is a program that performs tasks for its user"* and further continues that such agents should involve the notion of *trust, personalizability,* and *autonomy* [72]. In [22] the same author proposes additional desirable agent characteristics such as *ability to handle discourse, domain specifics, graceful degradation, cooperation, anthropomorphism, and taking care of the user's expectations*. The emphasis is clearly put on the social characteristics of agents allowing them to be *perceived as intelligent* by the user (see the Turing test [1, p. 5]).

3.6. Brustoloni agent

According to Brustoloni [23] *"Autonomous agents are systems capable of autonomous purposeful action in the real world"*. The requirement that Brustoloni's agents are able to act in the real world clearly distinguishes them from *software agents* (agents working in a purely software environment without a hardware architecture that they can directly manipulate).

3.7. Wooldridge and Jennings agent

Wooldridge and Jennings in [24] propose that there can be defined *two notions – weak and strong* – of characteristics that determine a software or hardware architecture as agent. The *weak notion* includes the properties of *autonomy, social ability, reactivity,* and *pro-activeness*. The *strong notion* adds to these characteristics the notion of *mental components* such as *believes, desires, intentions, knowledge*, etc. The authors propose their definition: *"an agent is a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives"*. As described in [25] the key concepts in this definition are *situatedness* (receiving sensory input and acting in the environment), *autonomy* (ability to act without direct human manipulation and having control over own internal state) and *flexibility* (social ability, responsiveness, and pro-activeness).

3.8.Franklin and Graesser agent

On the basis of the extensive review of existing definitions Franklin and Graesser propose their own definition of autonomous agent concept [10]: *"An autonomous agent is a system situated within and a part of an environment that senses the environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future"*. In addition to requirements for sensing and acting upon environment this definition states that the actions and perceptions should be closely connected and goal directed – actions should effect the future development of the processes in the environment that are sensed. There is no requirement the agents to be programs – they may be robots or humans. The authors demonstrate the capability of this definition to distinguish between some typical agent applications and conventional programs such as payroll program for example (it doesn't effect what it senses in the future, and it fails to satisfy temporal continuity requirement).

3.9. Luck and d'Inverno formalization of agent-related concepts

Based on the variety of existing definitions Luck and d'Inverno [25] advocate the need to define the agent related concepts more formally. They use Z specification language [26] in order to define several hierarchically interrelated concepts: *entity* – a set of attribute-value pairs describing certain things existing in the environment; *environment* - a collection of entities; *object* – an entity that can perform certain actions in the environment, entities with associated functionality; *agent* – an object that can be attributed a set of goals; and *autonomous agent* – an agent that has own agenda *"as opposed to functioning under the control of another agent"*.

The formal specification of these concepts and their properties and characteristics done in [25] can be viewed as a basis for more extensive formalization efforts needed to cover the great variety of existing agent-related applications.

4. Distinctive Features of Agents

In this section we will try to summarize the different characteristics (accepted with a different degree of consensus within agent research communities) that distinguish agents from other entities such as objects or software programs. A basic characteristic that virtually all researchers agree on is that agents are *situated in their environment* – perceiving it through sensors and manipulating it through effectors (see fig. 1). This is valid for real world (agents with hardware architecture), as well as for purely software type of environment (software agents). Usually between the processes of sensing and acting is introduced a middle layer, which main function is to provide appropriate mapping between the two processes. There are different agent architectures proposed – some of them attributing to this middle layer high-level information processing activities such as reasoning, planning or making decisions, other try to improve the effectiveness by employing direct relations between the perceptions and actions. May be most powerful is the combination of these two approaches.

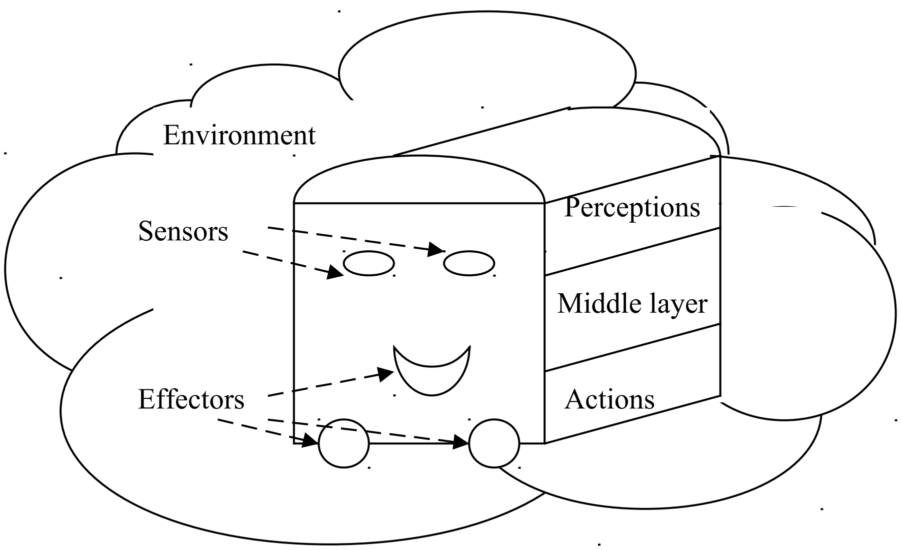

Fig. 1. Intelligent agent – basic architecture

The following characteristics should not be considered as means for excluding some existing applications from the class of "agent-based" ones, because probably no currently existing application implements all of them in full extent. Rather they provide a useful measure helping to determine where on the continuous (not binary) axis between traditional programs and ideal agents certain applications are.

4.1. Autonomy

In most definitions the autonomy is identified as central property of agents distinguishing them from traditional programs (e.g. text processor or payroll program). But as Jennings, Sycara and Wooldridge mention in [25]: *"Autonomy is a difficult concept to pin down precisely, but we mean it simply in the sense that the system should be able to act without the direct intervention of humans (or other agents), and that it should have control over its own actions and internal state"*. The agent's ability to have control over its behaviour (actions) clearly distinguishes it from *objects* in OOP which have control over their internal state (by encapsulating it) but not over the actions (messages, methods) that will be invoked (passed to them) by other external objects. In OPP it is designer's responsibility to assure that all method invocations are correct. In agents paradigm it is possible for an autonomous agent to decide that the required action is not feasible or acceptable from agent's point of view, and therefore should not be executed, or as summarized in [25]: *"Objects do it for free; agents do it for money"*.

Russell and Norvig state that in stronger sense: *"a system is autonomous to the extent that its behaviour is determined by its own experience"* [1]. Though as they mention, a fully autonomous system may not be effective because initially it will act randomly, without external guidance. So the amount of the agent's pre-built knowledge should be balanced (according to the unique environment requirements) in order to provide stable and effective initial behaviour, but not to make it rigid or limiting agent's flexibility and ability to adapt by learning from its own experience.

Luck and d'Inverno identify two possible views of autonomy [27]:

- *strong view of autonomy – "regarded as absolute without dimension or measure of degree"*;
- and *weak view of autonomy* – practically oriented, *"in which autonomy is taken to be the same as independence, a very distinctly relative notion"*.

They state further that the property of autonomy can be understood in terms of agent's *motivations* and *goals*: *"In essence, autonomous agents possess goals that are generated within rather then adopted from other agents. These goals are generated from motivations, higher-level non-derivative components characterizing the nature of the agent that can be regarded as any desires or preferences affecting the outcome of a given reasoning or behavioural task"*.

It is important to mention that the autonomy may not be appropriate or needed for all possible agent applications. The are areas where agents' ability to generate their own goals may be undesirable or even dangerous – e.g. live supporting or military applications. Murch and Johnson state in [11]: *"We think that it is important to add to the definition the restriction that agents work on behalf of others, that is, they are NOT self motivating. ... If they have their own goals, these goals are not part of their function as agents"*. In this way they completely exclude from agency the abovementioned strong view of autonomy.

4.2. Reactivity (Responsiveness)

According to [24] the agents are *responsive (reactive)* in sense that *"agents should perceive their environment and respond in a timely fashion to changes that occur in it"*. As states Wooldridge in [5, pp. 48-54] central for *reactive agents* are the ideas that: 1) intelligent behaviour is not disembodied (like in most expert systems) but *situated* in the environment – *"a product of the interaction the agent maintains with its environment"*; 2) *"intelligent behaviour emerges from the interaction of various simpler behaviours"*; 3) reactive agents should avoid using symbolic representations and syntactic manipulation in order decision process to be effective and efficient (see the "calculative rationality" problem described above). The approach is also called *"behavioural"* because the agent's decision making (finding appropriate actions in response to certain stimuli – see fig. 1) is centered around defining the simple behaviours needed to successfully accomplish the task.

The best known example of this type of reactive agent architectures is the Brooks *subsumption architecture* [28, 29, 30]. In this architecture the middle layer of symbolic knowledge representation (fig. 1) is removed and replaced by a collection of multiple concurrent behavioural patterns directly connecting perceptions with actions. In order to determine which of the behaviours triggered by a given situation will be executed a *subsumption hierarchy* is established in which different behaviours are prioritized. The low levels in this hierarchy represent more simple and concrete behaviours, while higher levels represent more general and abstract ones. The main prioritization rule is that the lower is the level in the subsumption hierarchy, the higher is the priority of the behaviour. This architecture has been effectively applied especially in the field of robotics, but there are several fundamental problems in front of purely reactive architectures connected with the "local" type of decisions they facilitate and the lack of knowledge explicitness compared with the symbolic approach (for more detailed discussion see [5, pp. 48-54]).

In an attempt to formally define properties of agents Goodwin [31] makes further distinction between *reactive* and *reflexive agents* with a main difference that: *"An agent is reflexive if it responds only to immediate stimulus. Such agents are also called stimulus-response agents. Reflexive agents don't need to maintain any memory …"*.

Of course there is no need all agents to be purely reactive – hybrid architectures may be developed that combine the strengths of reactive paradigm with other agent development mechanisms proposed.

4.3. Pro-activeness

By calling an agent proactive we mean that *"agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal directed behaviour and take the initiative where appropriate"*[4]. This type of agents are also called "deliberative" agents because in order to take the initiative and propose a viable solution to the user they should construct their own plan or strategy for task accomplishment. In order to make assumptions about *when* to propose *what*, these agents should be able to build and maintain a model of other participating agents (including the user) – their *believes, desires* and *intentions*. The BDI (believes, desires, intentions) theories are based on the theory of human practical reasoning proposed by Bratman [32] and have formal logic axiomatizations [33]. The main idea of BDI approach is that the *beliefs* an agent has determine its *desires*, or options it has to choose from, which in turn influence agent's *intentions*, or commitments made. The process is not linear because these commitments automatically change the set of available opportunities to choose from (remain only these opportunities that are compatible with the new made commitment – intention). For a detailed description of the BDI model see [34].

The pro-activeness and reactivity can be considered complementary to each other. A balance should be made between them depending on the characteristics of the task and the environment.

4.4. Social Ability (Communication)

While reactive and proactive agents represent the two extremes on the axis of agent interrelations, the *interacting* agents able to *communicate* with each other and showing mixed reactive/proactive behaviour provide much wealthier set of opportunities for carrying out complex tasks by building multiagent systems (MAS).

Looking on the history of human technological and cultural development we may see that most of the existing achievements would not be possible without the close interaction between many individuals and groups, and without the social conventions and language guiding this process. Nearly the same importance (and opportunities) some researchers see in finding appropriate interaction mechanisms and standard communication language for the multiagent societies. Genesereth and Ketchpel even state in [35] that a program *"is a software agent if, and only if, it communicates correctly in an agent communication language"*.

According to Huhns and Stephens in [5, pp. 79-120] there are three important characteristics of MAS: *"1) Multiagent systems provide an infrastructure specifying communication and interaction protocols;*

2) Multiagent environments are typically open and have no centralized designer; 3) Multiagent environments contain agents that are autonomous and distributed, and may be self-interested or cooperative". The ability to build open, multiagent systems capable of distributed task performance is providing several important advantages compared with more traditional centralized computing architectures:

- Combined with the global connectivity provided by Internet it gives rise of a new generation of complex, geographically distributed systems able to add new components dynamically and to be reconfigured in real time according to changes in the environment (information and communication environment, as well as business needs);
- Possibility for different vendors to build heterogeneous components (agents) that will be able to communicate, share information and knowledge with each other;
- Opportunity by using these technologies to cover wide scope problem domains – by joining the efforts of existing communities. This joined effort will be enabled by providing common standard knowledge representation formats, language and communication mechanisms.
- The intelligent distributed multiagent systems provide a paradigm for building scalable and robust software applications, where problems with continuously changing environment characteristics (e.g. network configuration, load, errors and exceptions) will be handled in a natural way. For example the load-balancing task can be accomplished by dynamically creating new agents and distributing them among the network nodes in order to efficiently use existing computational resources.
- The establishment of high level, task-oriented protocols and communication mechanisms will facilitate further development of existing software application areas such as e-commerce and e-business, e-learning, e-government, etc. It will exploit the theoretical insights about the architecture, interaction, and functioning of *self-interested (competitive)* or *cooperative* agents.

In order these opportunities to become reality, there is a need of common standards – especially a standard *agent communication language* (ACL) that all heterogeneous agents will be able to understand. According to [36] the abbreviation ACL has given two different meanings: 1) a general designations of all existing languages for multiagent communication; and 2) the name of a concrete communication language. The authors discuss that there have been proposed two major communication languages: ACL and FIPA ACL. The first of them originates in DARPA Knowledge Sharing Effort initiative started in 1990 by US Department of Defense. There was developed a conceptual framework for multiagent communication and knowledge sharing consisting of three main components: 1) *Knowledge Interchange Format (KIF)* [37] – a first order logic based format for knowledge representation corresponding to the *syntactical* aspect of communication; 2) *Ontology* representation languages (Ontolingua [38]) – providing a common vocabulary for the communication – *semantic* aspect; 3) *Knowledge Query and Manipulation Language (KQML)* [39, 40] – representing the agent's *propositional attitude* towards the information being communicated – corresponding to the *pragmatic* aspect of the communication.

The second major communication language – FIPA ACL – was proposed by Foundation for Intelligent Physical Agents (FIPA) [73]. FIPA is a non-commercial international organization developing standards in the area of agent-based systems and MAS. According to the FIPA Statement of Intent (available from [73]) the organization's core mission includes: *"The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings"*. FIPA ACL differs from above discussed ACL in that it does not make commitment to particular language for syntactical knowledge representation. The language is based on the notion of *"communicative acts" (CA)* corresponding to the KQML communication primitives (*performatives*) representing the propositional attitudes of agents. The semantic aspect of the communication in FIPA ACL is addressed by *SL* – a formal language for defining semantics. It represents a multimodal logic with modal operators for believes, desires, uncertain believes and intentions. The question which of the languages should be chosen for particular project has no obvious answer and depends on the project's requirements and the task characteristics. The perspectives in front of ACLs are connected with integration with other existing knowledge specification formalisms and languages like W3C's Extensible Markup Language (XML), Resource Description Framework (RDF), and DAML+OIL [74].

From architectural point of view the *social ability* realization leads to development of *multilayer agent architectures* [5, pp. 61-67]. These architectures can be *horizontal* – with each layer directly connected to perceptions and actions and providing a *mediator* function making decisions about which of the proposed behaviours to choose, and *vertical* in which only some of the layers are connected to perceptions and actions (see fig. 1). A good example of a vertically layered architecture is the INTERRAP system described in [41], which incorporates separate layers responsible for reactive, deliberative (proactive), and communicative (social) types of behaviour.

The four properties described so far constitute the Wooldridge and Jennings *week notion* of agentness. According to the authors they play central role in what most researchers designate by the term "agent". Following properties are not so well agreed and widely presented in all current agent architectures but they emphasize important aspects and directions for further development of the agent technology.

4.5. Rationality

Most authors understand by "rationality" the assumption that an agent will choose to perform those actions that according to agent's own predictions will achieve the goal defined, and will not choose those actions that according to the predictions will not achieve the goal. As Russell and Norvig state in the definition given in [1, p. 33] *"an ideal rational agent should do whatever action is expected to maximize its performance measure"*, according to the information (and knowledge) about the environment it has. The same authors mention that important component of agent's rationality is to choose to do actions needed *"to obtain useful information"* about the environment in order to complete the task successfully. The rationality is opposed to omniscience – we are not expecting the rational agent to do the *right thing*, but only to choose to perform these actions that he *is expecting* to achieve the goal. The chosen actions may be not optimal due to incomplete, insufficient, or inaccurate information or knowledge.

Goodwin provides more formal definition of rationality in [31] using *utility function* to define a measure of success of the agent. According to him: *"a utility based rational agent is one that prefers plans that have at least as high as minimum predicted utility in all possible starting conditions"*.

4.7. Benevolence and Veracity

Luck and d'Inverno [25] describe the property of *benevolence* as meaning *"that agents will cooperate with other agents whenever and wherever possible",* and *veracity* as *"not knowingly providing false information"*. As authors mention the *"blind benevolence"* is opposite to the concept of autonomy, because the autonomous agents are expected to choose only those actions that are considered advantageous according to their motivations. Though it is possible autonomous agent to *show* benevolent behaviour as a result of "selfish" motivations.

The benevolence and veracity properties can be viewed as connected with the *risk and trust* aspect of agent development described by Foner [22]: *"The idea of an agent is intimately tied up with the notion of delegation. We cannot delegate a task to someone or something else if we do not have at least a reasonable assurance that the entity to which we delegated can carry out the task we wanted, to our specifications"* and *"…we have to balance the risk that the agent will do something wrong with the trust that it will do it right"*. As main factors that determine the degree of user's trust he identifies 1) the completeness of knowledge (model) the user has about the agent's work, and 2) the importance of completing the task without failures.

4.8. Temporal Continuity

The temporal continuity property is usually understood in sense that the agents are not one shot computations (in distinction with most of the traditional software applications like text processors of payroll programs, which are started only when needed to accomplish their task and after that stopped) but rather they continuously monitor the environment and react on its changes. The temporal continuity can be considered as enabling other agent properties like autonomy, pro-activeness and learning. Each time agents have to complete given task they may handle it differently (unlike the traditional programs) due to their increased knowledge and experience.

4.9. Adaptability (Learning)

The ability to adapt to changes in the environment becomes crucial for agent's success when the environment is complex, dynamic (rapidly changing), and non-deterministic, which is the case for most of the existing real world and artificial multiagent environments. A variety of methods for *supervised* and *unsupervised* learning have been proposed by the AI community (ranging from inductive logic programming to artificial neural networks and genetic algorithms). The social ability property described above greatly facilitates the knowledge exchange between agents and therefore learning. Actually the learning is implicitly included in many of the agent models and approaches discussed so far (e.g. BDI theories and architectures).

Of course the knowledge exchange with other agents is not the only possible learning opportunity. The types of learning possible (and preferable) depend on the characteristics of the task and the environment. Maes identifies four general learning mechanisms suitable for a particular class of *"interface agents"* [42]:

- by observing and imitating user's behaviour;
- by adapting personal behaviour according to the received feedback from the user;
- learning from examples provided by the user;
- by receiving advice from other interface agents with more experience in accomplishing similar tasks.

A good overview of existing approaches for unsupervised learning suitable for autonomous agents can be found in [43].

4.10. Personality and Anthropomorphism

These agent properties are definitely not obligatory or even applicable to all existing types of agents. The advantage of showing a *believable personality* to the user is most obvious in the development of interface agents and avatars (virtual representatives of the user in a computer generated graphical world that the user can command – see the section about the types of agents and agent applications). It can provide the user with a useful graphical metaphor intuitively describing the agent's purpose, actions and internal state (including the agent's "emotional state" if such is modelled – e.g. by showing different face expressions and gestures).

The anthropomorphic graphical presentation of the agents can further reduce the user's cognitive load when learning to work in a new computer-based (and therefore artificial) environment. Of course in order these visual metaphors to be effective, they should be consistently supporting the user's expectations. It seems obvious that most people expect that anthropomorphic agents will behave (from the user's perspective) like other people – being able for example to communicate in natural language and to understand it.

As recognized by Foner [22]: *"there are several extent systems that might fit the 'agent' mold that are clearly not anthropomorphic (e.g., mail-sorting programs that learn how to sort based on watching the user's actions and making inferences), while there are some that clearly are (e.g., Julia). Hence, agency does not necessarily imply a need for anthropomorphism. Conversely, just because a program pretends to be anthropomorphic does not make it an agent"*.

The personality and anthropomorphism properties when consistently supported may help to increase the user's trust in the agent's capabilities, by helping the user to model more effectively the agent's work.

4.12. Mobility

Though being different, the agent characteristics described so far have many interconnections and dependencies. The *mobility* characteristic is exception – it can be vied as orthogonal (independent) to other properties. It is not enabling the realization of new agent functionality, but instead it provides new ways for more *efficient realization* of that functionality.

According to the Object Management Group (OMG) Mobile Agent System Interoperability Facility (MASIF) specification [75]: *"A mobile agent is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another"*. The main difference between *mobile* agents and *stationary* ones is that the former in addition to the ability to transport data using the computer network can transport their own code and internal state to a new location (execution environment).

Among the mobile agents advantages identified by Lange [44] are:

- *reducing the network load* – by transporting the computations to the host where the data resides and performing them locally;
- *overcoming the network latency* – especially important for real time process control systems, where the latency introduced by the computer network may be not acceptable;
- *increasing security by communication protocols encapsulation* – it may be important advantage to work with sensitive data only locally without transporting it over the networks;
- *ability for autonomous and asynchronous execution* – it is possible to start and instruct a mobile agent about the task parameters using one device (e.g. a mobile device) and to transfer this agent to the network host(s) where computations will be more efficiently executed, without a need for the mobile device to be permanently connected to the network;
- *load balancing through dynamic adaptation* – the agents can use the computational resources more efficiently by adapting their requirements to different conditions of the execution environments, and by choosing to migrate to a new host when the current host's resources are not sufficient;

- *facilitating building heterogeneous architectures* – by implementing the mobile agents in a hardware and transport protocols independent manner;
- *facilitating the implementation of robust and fault-tolerant systems* – the mobile agents can react dynamically to identified problems with given host by migrating to another host preserving their internal state and computational results already obtained.

The agent characteristics described above emphasize important aspects of the existing agent technologies, providing a basis for classification of different agent types and applications.

4. Types of Agents and Agent Applications

There are several dimensions that can be used to classify different types of agents and agent-based applications:

- Based on presence of specialized hardware architecture – the agents may be divided to *robots* maintaining a specific hardware architecture including specialized sensor and effector devices, and *software agents* or *softbots* (software robots) that are inhabiting a purely software execution environment. Some researchers state that this distinction may not be so important from architectural point of view, because there are existing really complex software environments (e.g. for interactive computer-based instruction), and some real world robot's tasks can be relatively simple (e.g. sorting details from a conveyor belt). [1]
- Based on software architecture – Russell and Norvig propose four basic types of agent architectures: *simple reflex agent* (reacting to sensor input by matching perceptions to actions using simple production rules), *reflex agent with internal state* (keeping track of the world), *goal-based agent* (employing planning algorithm in order to determine the sequence of steps needed for goal achievement), and *utility-based agent* (using utility function in order to combine multiple goals) [1]. Of course this is just one possible classification of agents' software architectures, and other classifications may also exist (e.g. horizontal, vertical, and hybrid layered architectures described above, or heterogeneous vs. homogeneous agent architectures).
- Based on AI methodologies applied – deliberative (using problem solving, decision making, learning, and planning algorithms) vs. reactive agents (employing a hierarchy of relatively simple behaviours directly connecting the percepts and actions).
- According to agents' communication ability – standalone, communicating, and communicating using ACL.
- According to mobility characteristic – stationary vs. mobile agents.
- According to visual representation – believable agents that are visually presented providing a visual metaphor of agent's purpose, goals, actions and state, vs. information processing agents that have to visual appearance.
- Based on the type of the environment agents inhabit – according to [1] there are different types of environments that can be classified using several binary properties like *accessibility* of the information about the environment's state (depending on what sensors the agent has), *dynamics* of the environment, *determinism* of the environment (are the changes in environment's state uniquely determined by the chosen agent's action), etc.
- Based on the purpose of agent-based applications – different application areas will be considered briefly in the next paragraphs.

There is a huge amount of agent-related projects and applications already existing and it is impossible to describe them in a single paper. That is why another approach is taken – we will attempt to describe the main application areas, giving examples when necessary. According to [4, 5, 11, 44, 45 84, 76, 77, 78] the following areas are identified as main candidates for agent-based technologies application:

- Information Management Applications – these applications will help users to manage information more effectively addressing the two main problems typically associated with Internet: information gathering and information filtering [4]. The current Internet architecture consists of two main layers: *suppliers* and *consumers* of goods, services, and information. The main way to find information in Internet now – by using existing *search engines* has several limitations [79] that can be addressed by introducing a middle layer of intelligent agents as matchmakers between the suppliers' offers and consumers' queries. Among the advantages of this approach are the ability to provide better matching by using cooperative agent processing (some agents can be experts in different problem domains) in order to handle complex quires, and possibilities to cope with the dynamic nature of the information in Internet – e.g. when a new supplier enters the market or existing one is changing the location of already offered resource, they can be sure that all customers will be able to find that resource by just sending a new offer to the matchmaking agents. According to [76] there are four major types of information management applications: directory services – yellow and white pages, data base enquiry, information brokerage, and media indexing.
- Interface Agents and User Assistant Applications – according to IBM report [45] there are several important factors that determine the need of new intelligent agent technologies in human-computer interaction (HCI) domain: the constantly rising complexity of computer hardware and software the users have to deal with; the growing quantity of information they have to process and filter; the users' increased mobility; and the need for making information technologies more acceptable for inexperienced computer users. Some examples of contemporary systems helping the users to cope with information overload are described in [46] and [47]. BotSpot is a website [80] dedicated particularly to different kinds of such agents (or "bots") helping to find different kinds of Internet resources in more effective and personalized way. The main desirable characteristics of this type of agents include the ability to personalize the user's experience by building and maintaining a *user model* (see the section about agents' adaptability and learning where different approaches for user modelling are discussed), to provide more *natural* and *user-centered* HCI approach incorporating *multimodal*communications by using speech synthesis and recognition, facial expressions and gestures. A typical scenarios for using this type of agents include the roles of *secretary* or *personal assistant*. They should be adaptable to user's preferences, should automate the repetitive tasks, and should serve as a communication centers integrating different information channels (e-mail, fax, telephone, audio and video conferencing, etc.) to facilitate the users' efficiency and mobility. Other possible tasks include scheduling user's appointments through automatic negotiations with other users' assistants (social ability), and multi-user collaboration support.
- Service Management Applications – according to [76] here are included the following services: *multimedia services* (interactive multimedia, video and television); *e-commerce buying/selling services*; *intelligent network management services*; and different concrete services such as *trip planning and guidance*, *ticket reservation*, etc. The important issues that should be addressed by this type of agent applications are the *quality of service (QoS), cost, time, security* and *privacy*. Kramer [81] states that: *"Mobile agents serve as a framework on top of which decentralized*

infrastructure services can be built. By embedding functionality in mobile software agents and distributing these agents across the network, we push the intelligence traditionally centralized in a few controlling nodes out into the system at large". This type of decentralized management is especially useful for building adaptive networks that will be able to effectively handle the dynamic configuration changes (e.g. in the case of mobile clients).

- E-Commerce Applications – several examples of existing agent-based e-commerce applications and architectures are described in [48, 49, 50] (to name just a few). The *Consumer Buying Behaviour Model* [51] identifies six stages in the *electronic marketplace* business process: 1) *need identification* (the consumer becomes aware of existing needs – goods, information or services); 2) *product brokering* (finding appropriate set of products that satisfy that needs); 3) *merchant brokering* (compiling a list of concurrent vendors based on a set of criteria: price, warranty, availability, delivery time, etc.); 4) *negotiation* (can be done automatically between the client and supplier agents in order to establish the optimal price and other purchasing conditions) 5) *purchase and delivery*; 6) *service and evaluation*. In each of these six stages the interface and information processing agents can be of great help – they can proactively identify the existing needs and opportunities and suggest them to the user, by communicating with other agents – domain specialists in the particular product area they can find out the desired product characteristics. After that by contacting matchmaking agents they will find the supplier agents to negotiate with. The purchase and delivery also can be automatically initiated and tracked including the payment process. And finally they will provide the user with appropriate (personalized and adaptive) information about the product, its maintenance, and troubleshooting. A discussion about the future of agent-based e-commerce applications can be found in [52].
- Business Process Management Applications – in addition to *electronic commerce* other opportunities for business processes automation using intelligent agents are proposed in [76] including: *financial services* (constantly monitoring stock information and financial news for relevant changes, giving advice to the user about prospective stocks, acting on his/her behalf by performing electronic transactions – an architecture for developing such agents called RETSINA is described in [53]); *workflow management* (coordination of business activities within a business organization and between business partners – e.g. supply chain management); *office automation* (document workflow management, personalized multimodal document presentations using avatar agent representing the document author [54]); *cooperative task management and groupware applications* (cooperative document authoring, group meeting facilitation); etc.
- Industrial Applications – according to [4] following areas are the best explored candidates for industrial scope agent-based applications: 1) *manufacturing management* [55, 56, 57, 76] – by applying different agent-based models for manufacturing scheduling optimization (e.g. Contract Net in [55]) and using flexible manufacturing systems (FMS); 2) *process control* (different applications range from climate control to particle accelerator processes control – see [4] for overview); 3) *telecommunications* – especially for large, mobile, dynamically reconfigurable telecommunication networks the intelligent agents can be needed in order to offer more flexible, secure and robust services [81]; 4) *air traffic control* – OASIS [58] was one of the successful large scale implementations of agent technology based on BDI approach; 5) *transportation systems* (agent-based traffic management and optimization) [4].
- Service Robotics Applications – in addition to industrial automation the agent-based robotics is automating services such as office mail delivery, package transportation, and vacuum cleaning (see the Roomba vacuum cleaner discussed above [70]). A top-down approach for agent-oriented

software engineering of cooperative robots is described in [59]. As proposed in [60] a tendency exists toward standardization of different robotic modules in order to be able to combine them in custom architectures according to different needs and to enable creation of market for reusable robotic components. The first such standard developed by Sony is called OPENR and it *"enables hot plug-in capability through precise definition of mechanical, electrical, and software inter face protocols"[60]*. Another initiative aimed at research of cooperative agent behaviour is RoboCup [61, 82]. It's ultimate goal is in 2050 a team of autonomous agents to be able to play soccer against the human champions team. There are two leagues in RoboCup – for hardware and for software robots playing soccer. The championships are organized in a regular basis several times yearly. Another goal of the RoboCup organization is to use the experience accumulated in constructing teams of soccer players for development of cooperative robots for rescuing people in natural disasters – the project is called RoboCupRescue.

- Software Application Development – as described in [62] the agent-based approach becomes a viable alternative of well known object-oriented development techniques, providing several important advantages: 1) ability to change the system configuration and functionality dynamically by adding and removing appropriate agents; 2) building systems that are more robust and fault tolerant; 3) ability to build complex, distributed, and heterogeneous systems; 4) enabling the development of self-organizing and adaptive systems by using multiple cooperating agents; 5) agents are able to work in parallel (possibly on different computers) improving the overall performance of the system. Actually the object-oriented and agent-oriented approaches are not alternatives (the agents can be viewed as active objects [8]) – they can be combined in order to build complex real-time applications.
- Medical and Healthcare Applications – according to Murch and Johnson [11] there are several categories of medical processes connected with diagnosis and treatment of diseases: *diagnosis, treatment, recuperation, follow-up, administration, education*. Several scenarios may be proposed for automating each phase. An important problem with patients having chronic health problems is the need for continuous monitoring of their state (blood pressure, pulse, etc.). It is inconvenient and expensive to organize continuous observation of the patient's state by medical staff accessible 24 hours a day. An autonomous agent(s) can introduce improvements in several directions: continuous monitoring of all the parameters of the patient's state; 2) ability to predict possible crisis by recognizing different patterns of input parameters; 3) ability for the medical specialists to monitor the patient's state distantly (possibly through Internet) offering the patient less stressing and more comfortable rehabilitation conditions at home instead in the hospital (with the advantage of continuous monitoring); 4) possibility to collect a huge amount of information about the histories of different diseases that can be used to find the most appropriate treatment for the particular patient's conditions as well as for early diagnosis – e.g. by employing data-mining techniques. Hayes-Roth et at. describe in [21, 63] an ICU (intensive care unit) patient monitoring system called GUARDIAN. The key requirements to the system are that it *"must adapt several key aspects of its behaviour to its dynamic situation: its perceptual strategy, its control mode, its choices of reasoning tasks to perform, its choices of reasoning methods for performing those tasks, and its meta-control strategy for global coordination of all of its behaviour"* [21]. The agent dynamically constructs control plans by choosing *"among situation triggered behaviours"*. Huang et al. propose a prototypical health care system using intelligent agents [64].
- Entertainment Applications – as Maes states in [20]: *"Entertainment is an extremely large industry that is only expected to grow in the near future. Many forms of entertainment employ*

characters that act in some environment. This is the case for video games, simulation rides, movies, animation, animatronics, theater, puppetry, certain toys and even party lines. Each of these entertainment forms could potentially benefit from the casting of autonomous semi-intelligent agents as entertaining characters. Entertainment is a fun and very challenging application area which will push the limits of agent research". In the same publication are described several applications ranging from *believable agents* – autonomous animated characters that posses some built-in behaviours and can perceive and react to each other generating more interesting animation movies that if produced by traditional "static" method (it depends on how realistic and consistent these programmed behaviours are), through conversational agents (like Foner's Julia [22]) that inhabit a text based multi-user simulation environment, to Artificial Life Interactive Video Environment (ALIVE) project developing *"a virtual environment which allows wireless full-body interaction between a human participant and a virtual world which is inhabited by animated autonomous agents"*[65]. Another type of agents with high potential for building entertainment applications are *avatars* [11, pp. 169-173]. The avatar is a virtual representative of a person in a computer-generated (usually three dimensional) world. The users can communicate with each other by controlling their avatars appearance – position, facial expressions, gestures, voices and sound effects. They can also talk to each other (e.g. using a "balloon-stile" chat) and show emotional expressions (for example showing lack of interest by increasing the distance with another avatar or by turning the face in different direction). Most of the avatar presentation systems offer to the user just a limited set of pre-programmed commands that he/she can use in order to change its avatar mood and behaviour. Recently more sophisticated avatars were developed using autonomous agent technology [66] capable of autonomous and realistic behaviour (by automatically animating certain behaviours such as attention expression, salutations, turn taking, facial expressions, etc.). The avatar-based chat rooms are not the only possible application of this technology – among the other applications are: participation in games (stand-alone and multiplayer), e-commerce applications allowing the user to try the products (cloths, shoes, etc.) and to see the result immediately on his/her avatar [11], the avatars can be attached to different documents helping the reader to understand the information and possibly answering to questions connected with the document [54], as personal assistants, news commentators, spokespersons, product representatives (avatar-based advertising), movie actors, etc. [68]. The intelligent agents can be used in film/video production – as autonomous camera agents, or graphical agents for story board design [76].

There are several emerging new areas of intelligent agents' application including (according to [78]) the notion of *ambient intelligence* – mainly introduced by the efforts of the European Commission in identification of research directions leading to Information Society Technologies development. The ambient intelligence (AmI) is representing the vision of building large-scale heterogeneous networks of interconnected intelligent devices able to support emerging *virtual organizations*. Among the main challenges in front of realization of this vision are the *distributed and dynamically changing configurations* of the network, its *heterogeneity* leading to the need of learning and adaptation mechanisms to be supported in order new types of such embedded devices to be accommodated seamlessly in the existing network (probably by using extensive ontologies describing their properties), the need for *scalability* (ability to scale the devices configurations dynamically), and *layered architecture* allowing to separate the different aspects of the interactions between distributed devices and services like *physical communication infrastructure*, *logical connectivity*, *resource discovery*, *access to information repositories*, etc. The intelligent agents technologies can be considered as main candidate for realization of this vision because of their inherent distributed nature, mobility, reactivity, pro-activeness, learning and adaptation to changing environment characteristics, and because of their communication and social interaction ability.

Another area of emerging intelligent agents applications is the *grid computing* – starting from building large scare distributed information processing systems its current tendency is to move towards *layered infrastructures of distributed services* that can naturally be implemented by exploiting the agent-based development approach .

Biology and Bioinformatics need technologies able to cope with the huge amount of information, knowledge, and computations produced in these areas. There is a need to effectively accumulate, locate, filter and represent the distributed knowledge that can be effectively done by a combination of intelligent agents technologies with other leading information processing technologies like datawarehousing and datamining.

The *e-learning* is another possible application area of agent technologies, that has been already partially explored by different researchers (for an example see [69]), but still offers many unresolved questions and problems as well as opportunities for building industrial quality interoperable learning management systems (LMS), learning content management systems (LCMS), and intelligent run-time learning environments. The interface and information processing agents discussed so far can help to satisfy this need and to overcome existing problems (inflexible user interactions, closed systems unable to exchange learning content, multiple standards and specifications complicating the task of building interoperable systems).

5. Conclusions

The intelligent agents technologies are currently one of the most promising and fast developing areas of multidisciplinary research. Ranging from personal assistance and entertainment to large industrial projects and robotics its applications will have strong influence over the evolution of technologies and human society. The rapid technological development in some cases outstrips the abilities of the society to find the most effective ways of using it. There are many technical, psychological and philosophical questions that should be answered in order intelligent agents to become part of our everyday live.

References

1. Russell, S., Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995
2. Newell, A., Simon, H., *GPS: A program that simulates human thought*, In Lernende Automaten, R. Oldenbourg, KG, 1961
3. Russel, S., Subramanian, D., *Provably bounded-optimal agents*, Journal of AI Research, vol. 2, pp. 575-609, 1995
4. Jennings, N., Sycara, K., Wooldridge, M., *A roadmap of agent research and development*, Autonomous Agents and Multi-Agent Systems, 1 (1), 1998, pp.7-38
5. Weiss, G. (editor), *Multiagent Systems: A Modern Approach to Distributed Intelligence*, MIT Press, 2000
6. Pressman, R., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 1997, pp. 567-596
7. Arrington, C., *Enterprise JAVA ™ with UML ™*, John Wiley, 2001
8. Booch, G., *Object-oriented analysis and design with applications*, Benjamin Cummings, 1994, pp. 27-77
9. Black, A., Hutchinson, N., Jul, E., Levy, H., Carter, L., *Distribution and Abstract Types in Emerald*, Report 86-02-04, Seattle, WA: University of Washington, 1986, p.3
10. Franklin, S., Graesser, A., *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, 1996
11. Murch, R., Johnson, T., *Intelligent Software Agents*, Prentice Hall, 1999
12. Longman Dictionary of English Language, 1988
13. Charniak, E., McDermott, D., *Introduction to Artificial Intelligence*, Addison-Wesley, 1985

14. Bellman, R., *An Introduction to Artificial Intelligence: Can Computers Think?*, Boyd & Fraser, San Francisco, 1978
15. Haugeland, J., editor, *Artificial Intelligence: The Very Idea*, MIT Press, 1985
16. Kurzweil, R., *The Age of Intelligent Machines*, MIT Press, 1990
17. Schalkoff, R., *Artificial Intelligence: An Engineering Approach*, McGraw-Hill, 1990
18. Winston, P., *Artificial Intelligence*, Addison-Wesley, 1992
19. Luger, G., Stubblefield, W., *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Benjamin/Cummings, 1993
20. Maes, P., *Artificial Life Meets Entertainment: Life like Autonomous Agents*, Communications of the ACM, 38, 11, 1995, pp. 108-114
21. Hayes-Roth, B., *An architecture for adaptive intelligent systems*, Artificial Intelligence: Special Issue on Agents and Interactivity, 72, 1995, pp. 329-365
22. Foner, L., *What's an Agent, Anyway? A Sociological Case Study*, in the proceedings of the First International Conference on Autonomous Agents (AA '97), Marina del Rey, California, February 5-8, 1997
23. Brustoloni, J., *Autonomous Agents: Characterization and Requirements*, Technical Report CMU-CS-91-204, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, November 1991
24. Wooldridge, M., Jennings, N., *Intelligent agents: Theory and practice*, The Knowledge Engineering Review, vol. 10 (2), 1995, pp. 115-152
25. Luck, M., d'Inverno, M., *A Conceptual Framework for Agent Definition and Development*, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), 1995
26. Spivey, J., *The Z Notation: A Reference Manual*, Prentice Hall, 2nd edition, 1992
27. Luck, M., d'Inverno, M., *Autonomy: A Nice Idea in Theory*, Lecture Notes in Artificial Intelligence, Springer, 2001, pp. 351-353
28. Brooks, R., *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, 2(1), 1986, pp. 14-23
29. Brooks, R., *Elephants don't play chess*, In P. Maes, editor, Designing Autonomous Agents, MIT Press, 1990, pp. 3-15
30. Brooks, R., *Intelligence without representation*, Artificial Intelligence, 47, 1991, pp. 139-159
31. Goodwin, R., *Formalizing properties of agents*, Technical Report CMU-CS-93-159, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1993
32. Bratman, M., *Intentions, Plans, and Practical Reason*, Harvard University Press: Cambridge, MA, 1987
33. Rao, A., Georgeff, M., *A model-theoretic approach to the verification of situated reasoning systems*, In proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), 1993, pp. 318-324
34. Bratman, M., Israel, D., Pollack, M., *Plans and resource-bounded practical reasoning*, Computational Intelligence, 4, 1988, pp. 349-355
35. Genesereth, M., Ketchpel, S., *Software Agents*, Communications of the ACM **37** (7), 1994, pp. 48-53
36. Labrou, Y., Finin, T., Peng, Y., Agent Communication Languages: The Current Landscape, IEEE Intelligent Systems, 14 (2), March/April 1999, pp. 45-52
37. Genesereth, M., et al., *Knowledge Interchange Format, Version 3.0 Reference Manual*, tech. report, Computer Science Department, Stanford University, 1992
38. Farquhar, A., Fikes, R., Rice, J., The Ontolingua Server: A Tool for Collaborative Ontology Construction, tech. report KSL-96-26, Stanford Knowledge Systems Laboratory, Stanford, 1996
39. *Specification of the KQML Agent Communication Language*, tech. report, DARPA Knowledge Sharing Initiative, External Interfaces Working Group, 1993
40. Labrou, Y., Finin, T., A Proposal for a New KQML Specification, tech. report, TR-CS-97-03, Computer Science and Electrical Engineering Dept., Univ. of Maryland, Baltimore, 1997
41. Muller, J., *A cooperation model for autonomous agents*, In Muller, Wooldridge and Jennings, editors, Intelligent Agents III (LNAI Volume 1193), Springer-Verlag, 1997, pp. 245-260
42. Maes, P., *Agents that Reduce Work and Information Overload*, In: Software Agents, edited by Jeffrey M. Bradshaw, AAAI Press/MIT Press, 1997
43. Maes, P., *Modeling Adaptive Autonomous Agents*, In: Artificial Life, An Overview, edited by Christopher G. Langton, MIT Press, Cambridge, 1995

44. Lange, D., *Mobile Objects and Mobile Agents: The Future of Distributed Computing?*, In Proceedings of The European Conference on Object-Oriented Programming (ECOOP'98), 1998, pp. 1-12
45. Gilbert, D., Aparicio, M., Atkinson, B., Brady, S., Ciccarino, J., Grosof, B., O'Connor, P., Osisek, D., Pritko, S., Spagna, R., Wilson, L., The Role of Intelligent Agents in the Information Infrastructure, IBM Report, 1995
46. Chen, L., Sycara, K., *WebMate : A Personal Agent for Browsing and Searching*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 132-139
47. Barrett, R., Maglio, P., Kellem, D., *WBI: a confederation of agents that personalize the Web*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'99), 1999, pp. 496-499
48. Krulwich, B., *The BargainFinder Agent: Comparison Price Shoping on the Internet*, In J. Williams, editor, Bots and other Internet Beasties, Macmillan, 1996, pp. 257-263
49. Doorenbos, R., Etzioni, O., Weld, D., *A scalable Comparison-Shopping Agent for the World Wide Web*, In Proceedings of the International Conference on Autonomous Agents (AGENTS'97), 1997, pp. 39-48
50. Barbuceanu, M., Lo, W., *A multi-attribute utility theoretic negotiation architecture for electronic commerce*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'2000), 2000, pp. 239-246
51. Nwana, H., Rosenschein, J., Sandholm, T., Sierra, C., Maes, P., Gutmann, R., *Agent-Mediated Electronic Commerce: Issues, Challenges and some*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 189-196
52. Terpsidis, I., Moukas, A., Pergioudakis, B., Doukidis, G., Maes, P., *The Potential of Electronic Commerce in Re-Engineering Customer-Retail Relationships through Intelligent Agents*, J. Roger et al. (editors), Advances in Information Technologies: The Business Challenge, IOS Press, 1997
53. Sycara, K., Pannu, A., *The RETSINA multiagent system: towards integrating planning, execution and information gathering*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 350-351
54. Bickmore, T., Cook, L., Churchill, E., Sullivan, J., *Animated autonomous personal representatives*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 8-15
55. Parunak, H., *Manufacturing Experience with the Contract Net*, In M. Huhns, editor, Distributed Artificial Intelligence, Pitman Publishing, 1987, pp. 285-310
56. Chung, K., Wu, C., *Dynamic Scheduling with Intelligent Agents: An Application Note*, Metra Application Note 105, Metra, Palo Alto, CA, 1997
57. Shen, W., Norrie, D., *Combining mediation and bidding mechanisms for agent-based manufacturing scheduling*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 469-470
58. Ljunberg, M., Lucas, A., *The OASIS Air Traffic Management System*, In Proceedings of the Pacific Rim International Conference on AI (PRICAI'92), Seoul, Korea, 1992
59. DeLoach, S., Matson, E., Li, Y., *Applying agent Oriented Software Engineering to Cooperative Robotics*, In Proceedings of Florida Artificial Intelligence Research Society Conference (FLAIRS 2002), 2002, pp. 391-396
60. Fujita, M., Kitano, H., Kageyarna, K., *Reconfigurable Physical Agents*, In Proceedings of the International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 54-61
61. Veloso, M., Stone, P., Han, K., *The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 78-85
62. Conrad, S, Turker, C., Saake, G., *Towards agent-oriented specification of information systems*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'99), 1999, pp. 502-503
63. Hayes-Roth, B., Washington, R., Ash, D., Collinot, A., Vina, A., Seiver, A., *Guardian: A prototype intensive care monitoring agent*, Journal of Artificial Intelligence and Medicine, 1992
64. Huang, J., Jennings, N., Fox, J., *An Agent-based Approach to Health Care Management*, Applied Artificial Intelligence: An International Journal, Taylor & Francis London, 9 (4) 1995,pp.401-420
65. Maes, P., Darrell, T., Blumberg, B., Pentland, A., *The ALIVE system: Full-body Interaction with Autonomous Agents*, Proceedings of the Computer Animation '95 Conference, Geneva, Switzerland, IEEE-Press, April 1995
66. Vilhjalmsson, H., Cassell, J., *BodyChat: autonomous communicative behaviours in avatars*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 269-276
67. LoPiccolo, P., *What Are 'Veople' for?*, Computer Graphics World, November 2002, pp. 4-6

68. Price, R., Douther, C., Jack, M., *An investigation of the effectiveness of choreography for the portrayal of mood in virtual environments*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'2000), 2000, pp. 54-55
69. Rickel, J., Johnson, W., *STEVE: a pedagogical agent for virtual reality*, In Proceedings of The International Conference on Autonomous Agents (AGENTS'98), 1998, pp. 332-333
70. iRobot website, available at: http://www.irobot.com, last visited 30 May 2003
71. Coen, M., *Software Agents*, available at: http://www.ai.mit.edu/people/mhcoen/agents/chapter2_1.html, last visited 30 May 2003
72. Foner, L., *What's an Agent?*, available at: http://foner.www.media.mit.edu/people/foner/agents.html, last visited 30 May 2003
73. FIPA (Foundation for Intelligent Physical Agents) official website, available at: http://www.fipa.org/, last visited 30 May 2003
74. World Wide Web Consortium (W3C) official web site, available at: http://www.w3c.org, last visited 30 May 2003
75. Object Management Group (OMG) official web site, available at: http://www.omg.org, last visited 30 May 2003
76. Azarmi, N., Leger, A., Kim, Y., Steiner, D., *FIPA applications types (Rev. 2.0)*, Yorktown, 1996, available at: http://leonardo.telecomitalialab.com/fipa/yorktown/nyapplications.htm, last visited 30 May 2003
77. AgentLink – Europe's Network of Excellence for Agent-based Computing website, available at: http://www.agentlink.org/resources/clearing-house-view.html, last visited 30 May 2003
78. Luck, M., McBurney, P., Preist, C., *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*, AgentLink, 2003, available at: http://www.agentlink.org/resources/clearing-house/RA5.php, last visited 30 May 2003
79. Hermans, B., *Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society & a prediction of (near-)future developments*, (thesis), Tilburg University, Tilburg, The Netherlands, 1996, available at: http://www.hermans.org/agents/, last visited 30 May 2003
80. BotSpot website, available at: http://www.botspot.com/, last visited 30 May 2003
81. Kramer, K., Minar, N., Maes, P., *Tutorial: Mobile Software Agents for Dynamic Routing*, Mobile Computing and Communications Review, 3 (2), March 1999, available at: http://www.media.mit.edu/~nelson/research/routes/, last visited 30 May 2003
82. RoboCup web site, available at: http://www.robocup.org/, last visited 30 May 2003

