Sapient Agents – Seven Approaches

Zbigniew Skolicki
Computer Science Department
George Mason University
4400 University Drive, MSN 4A5
Fairfax, VA 22030
phone: 703 993 3919
email: zskolick@gmu.edu

Tomasz Arciszewski
Civil Environmental and Infrastructure Engineering Dept.
George Mason University
4400 University Drive, MSN 4A6
Fairfax, VA 22030
phone: 703 993 1513
email: tarcisze@gmu.edu

Abstract — *During the last several years, a large class of computer programs, called " Intelligent Agents" (IAs), has emerged as a result of intensive research efforts in various countries. There is ongoing discussion how to classify the individual types of intelligent agents and how to distinguish between them and a subclass called "Sapient Agents" (SAs). In this paper, seven approaches are proposed to distinguish SAs from the entire class of IAs. These approaches are proposed mostly in the context of IAs for engineering design, but the results should have more general implications. The proposed approaches are based first on analyzing knowledge representations and the concept of emergence. Considering exploitation versus explorations in design and in evolutionary computations gives further approaches. Next, we consider SAs in the time and domain context, and then in the context of the chaos theory. Finally, initial conclusions and plans for further research are presented.*

1. Introduction

Engineering is undergoing a transformation mostly driven by the ongoing Information Technology Revolution, understood here as a complex process of interrelated changes in the areas of Computer Science, particularly in Artificial Intelligence and computing power.. One of the emerging technologies, with a potentially significant impact on engineering, is that of Intelligent Agents (IAs). In this paper, an IA is understood as *an autonomous system situated within an environment, which senses its environment, maintains some knowledge and learns upon obtaining new data and, finally, which acts in pursuit of its own agenda to achieve its goals, possibly influencing the environment* [13]. However, this descriptive definition covers an entire spectrum of agents of various complexities, behavior, and intelligence. At one end of this spectrum are simple homogeneous agents, like in swarm [9] while at the other end such sophisticated agents like Disciple [15] can be found.

When engineering applications of IAs are concerned, there is some confusion about the nature of intelligent agents and their usefulness for various engineering tasks. Also, it is obvious that a function of an agent that performs recording and classification of highway traffic is much different to a function of a complex agent conducting conceptual and detailed engineering design. Intuitively, it is natural to assume that some IAs can be classified as special, or "Sapient Agents," (SA) since their complex and difficult to predict behavior can be attributed to their "wisdom," which is different from knowledge in the form of a simple collection of decision rules, driving other, less sophisticated agents. SAs can be described in terms of their knowledge, or wisdom, but the authors believe that such an approach is simply insufficient. Therefore, they attempt to define SAs using a classical approach called "a definition by coverings." In this case, a concept, here a class of SAs, is defined by a number of descriptors representing (covering) various aspects (features) of a given concept. In the paper, seven major aspects of SAs, or their behavior, are discussed and proposed to identify (define) the concept of a SA. The presented approaches can be used separately, or eventually jointly, when our understanding of sapient agents is sufficiently improved.

2. Seven Approaches

2.1. Knowledge Representation

A distinction between a general class of IAs and SAs can be made considering knowledge representation. This can be done for the most popular forms of knowledge representation, i.e. for knowledge in the form of decision rules and knowledge represented as a graph.

In the first case of decision rules, it can be assumed that a SA has knowledge in the form of a knowledge system containing a collection of decision rules, meta-rules, models, heuristics, etc. By contrast, an "ordinary" IA has knowledge exclusively in the form of a collection of decision rules, which obviously is much more brittle and inflexible than in the case of a sapient agent.

In the second case of graph knowledge representation, one can easily imagine an IA analyzing the detailed relations between entities. An agent may use an ontology (which is in

the form of a graph) for the task of generalization/specialization of particular rules, as it is in [15]. However, a wise agent should probably understand the whole structure of a graph, abstracting from particular, single connections. The argument can be made that intelligent behavior would consider nodes representing detailed solutions, by analyzing their interrelation, whereas sapient behavior would rather look at interconnections between different parts of the graph. SA could use the topology of a meta-graph, where nodes are the sub-graphs of the knowledge graph, or use a separate structure representing the general features of the world modeled. In this context, **a SA can be defined as an IA capable of understanding the whole structure of a graph and capable of abstracting knowledge from this graph.**

2.2. Emergence

The phenomenon of emergence is understood as a sudden and unexpected occurrence of a new engineering pattern during an evolutionary process [1]. As an example, an evolutionary conceptual structural design process is used, first controlled by a human designer and next by an IA. During such a process, an entire line of evolution of structural design concepts is generated. It may happen that at a certain stage the human designer observes that the generated structural configuration is unknown, but feasible, better than known configurations, and even potentially patentable. In such a case, a new structural shaping pattern has emerged.

When, hypothetically, an IA controls the same process, an "ordinary agent" will be able to make the decisions regarding its continuation or termination, based on a set of assumed criteria being used in a mechanistic way. However, when a SA controls the process, it will be able to recognize the emergent pattern and stop the process, if desired. In this context, **a SA can be defined as an intelligent agent capable of recognizing emergent patterns.**

2.3. Exploitation/exploration

In design science [6], there is a clear distinction between exploitation and exploration in the context of conceptual design and searching a design representation space. If we assume that in the future the conceptual design process will be conducted by intelligent agents [7], as it is expected, then this distinction can be used to distinguish a SA from an IA.

In the case of exploitation, design concepts are sought within a known design representation space, which remains unchanged during the entire design process. However, when design exploration is conducted, an entirely different paradigm is used. In this case, the design representation space is supposed to be modified and expanded to allow the designer to find novel design concepts. There is no question that exploration is much more challenging than exploitation and that it requires an entirely different set of methods and tools. In this context, **a SA can be defined as an IA capable of conducting exploration while an "ordinary" IA is capable only of performing exploitation.**

2.4. Evolutionary Computation

Evolutionary Computation (EC) may be seen as a guided search mechanism, in which we make decisions based on the outcome of the previous decisions. In this way, EC finds the solutions much faster, because it is able to climb up a hill in the search space. However, the price paid for that ability is the danger of ending up in the wrong region of the search space and without any means for escaping from there.

EC may be seen as a compromise between exploration and exploitation [5]. High level of exploration makes EC closer to the Monte Carlo (random search) method. In this case, the high probability of finding the best solution is associated with uncertainty about the time required to find it. Random search, which blindly generates solutions under conditions of unlimited time, would finally find the optimal solution. Unfortunately, when engineering applications are concerned, usually only a limited amount of time is available, and thus only a limited number of points in the solution space can be analyzed. A high level of exploitation makes the algorithm climb up local hills and "zoom in" on the optimal solution. In addition, making the algorithm utilize local information increases the probability that it will find only a local optimum, a solution that may be much worse than the global optimum.

Several decades of research on EC have demonstrated that an algorithm looking only at current individuals in a given population, which are all uniformly driven toward a solution, may have significant difficulties in finding the optimal solution in complex domains. Researchers have learned that although the current situation considered on a local, or tactical level, would suggest going directly to a particular optimal solution, it is much wiser "not to hurry" and let the algorithm slowly drift toward the optima. The local evaluation, conducted at every generation, may not justify this approach. However, a general knowledge about the domain and the behavior of evolutionary algorithms suggests that locally unjustified steps, like choosing an individual with a worse fitness may produce a better outcome in the long term. Some counterintuitive (at least at first sight) actions are thus taken. It is very important to maintain the proper diversity of a population. It occurs that once we lose the wide spectrum of chromosomes, it is difficult to recover chromosome diversity. There are many approaches developed that aim to oppose fast converging. . They include multi-population EAs, Pareto optimization, sentinels, fitness sharing, niches and demes, tags, and others [3].

It is the authors' opinion that the methods mentioned above represent a wise approach to EC. They are based on experience with many evolutionary algorithms and provide long-term benefits. The behavior they dictate is not based on any locally available information, as opposed to intelligent behavior (at least according to our definition), and in some case may even seem unintelligent. In this context, **a SA can be defined as an IA capable of making strategic decisions, which may not be entirely justified by the local results.**

2. 5. Time

Although local decisions based on simple rules and utilizing available information may lead to temporary correct moves, the strategic moves are usually driven by long-term, general rules or plans. Such plans may not necessarily rely on actual, tactical knowledge, but rather on a higher-level wisdom (strategic knowledge) about how the environment behaves over long time.

Short term vs. long term goals contradiction is especially visible in the area of planning [12]. To facilitate the task of finding the proper order for a large set of actions, one utilizes the concept of hierarchical planning. In this approach, a general plan is constructed first and then the detailed plan is searched within the search space defined by the general plan. This mechanism, although it may miss some of possible solutions, discourages the system from expanding the local search too much in order to avoid the risk of loosing the global perspective and ultimately of missing the global optimum.

Heuristics implemented in computers playing different games, like chess [4], may be seen as another example of a long term, wise behavior. Although a local game tree search may suggest a different move, it seems sensible to implement some higher-level analysis of the game, which would establish a general plan of the game. Maybe that is why the game of Go was not mastered by computers yet [11] – computers still lack the holistic understanding (or feeling) of the game, which human players possess. It is interesting how often wisdom is related to intuition – a seemingly non-deductive knowledge, based on experience. In this context, **SA can be defined as an IA capable of making long-term decisions.**

2. 6. Domain Dependence

Information coming from a given domain, after generalization, creates a body of knowledge about this domain. What can be said, however, in general about the behavior of a given agent, regardless of the domain? Unfortunately, all domain-specific information cannot be transferred into another domain. Only general behavioral patterns remain valid and create a body of meta-level knowledge about possible relationships among entities, which is applicable to a number of various domains. Ultimately, abstraction of knowledge from different domains may produce high-level knowledge, or wisdom, which will be a basis for sapient behavior.

Similarly, knowing the general knowledge classification rules may help in acquiring new knowledge. If the domains are of a similar nature, one can suppose that knowledge about any new domain can be at least partially acquired using the knowledge structures developed so far for similar domains [10]. For example, human learning of foreign languages can be considered. Such "knowledge about knowledge" can be quite abstract, as shown in [14] using the example of three basic entities of object, supplement, and a relation (or Firstness, Secondness and Thirdness). Also, it is important to have proper intuition (or a collection of heuristics) when abstracting knowledge related to a particular domain. This knowledge should suggest which entities are more important and can later be expanded into many more detailed categories, and which entities are less important and will most probably remain unchanged. Specifically, such intuition (or meta-knowledge) may help to properly build ontologies and avoid their later restructuring. Currently only human experts are able to address this issue. However, one can hope that in the future SAs will be capable of making predictions about the nature of knowledge to be acquired. Finally, a SA should be able to adapt its own behavior in the knowledge acquisition process depending on the results of its own actions. Although this ability is to some extent a feature of all IAs - because they all actively search for new information - agents that modify the behavior at a very deep level, for example completely changing the way they store knowledge, should definitely be considered as SAs.

In the context of the above remarks, **a SA can be defined as an intelligent agent capable of abstracting knowledge and of adapting its behavior.**

2.7. Chaos Approach

One of the most important goals in the analysis of chaotic systems is the identification of their attractors. When a given representation space contains feasible design solutions, the attractors represent the locally best solutions [2]. In this case, two very similar starting design solutions (points in the representation space) may lead to completely different final solutions, although they are both obtained by deterministic, algorithmic means. This observation may suggest that a proper selection of initial solutions could lead to a better final solution. Alternatively, after converging to a solution situated on a particular attractor, one could try to restart the search process to see if there is another attractor representing a much better solution.

A design process can be considered as traversing one large search space. Then attractors correspond to various final solutions and other points in the space represent solutions obtained in the earlier stages of the design process. In such a case, it is obvious that two similar initial conceptual

Table 1 Comparison of IAs and SAs.

No.	Definition Name	Intelligent Agent	Sapient Agent
1	**Knowledge Representation**	Only decision rules	A knowledge system
2	**Emergence**	No recognition of emerging patterns	Recognition of emerging patterns
3	**Exploitation/ Exploration**	Conducts only exploitation	Capable of conducting exploration
4	**Evolutionary Computation**	Classic Evolutionary Algorithm (tactical decisions driven by current population)	Algorithm maintaining diversity, long-term benefits in complex and dynamical environments (strategic decisions driven by global understanding)
5	**Time**	Capable of making only short-term decisions	Capable of making long-term decisions
6	**Domain Dependence**	Limited adaptive behavior	Capable of abstracting knowledge and of adaptive behavior
7	**Chaos**	Unaware of attractors	Capable of avoiding or finding attractors

designs, located close to each other in the space, may in the search process result in two different final solutions, having entirely different final features. Therefore, a SA should probably not focus on a single particular design from the very beginning, but "keep in mind" other possibilities. Such an approach would prevent big changes later in the design process, allowing for a proper adjustment early enough. Of course, it may not be clear, or even it may be impossible at the conceptual stage of the project, to predict which particular way of solving a problem is better. It may be that the later, detailed analysis reveals some constraints impossible to foresee by the designer. Therefore, again, some kind of general knowledge about domains, some "engineering intuition," is needed to choose the right path, or to restrict thousands of possibilities to the very few important ones.

Chaotic dynamical systems, except of having the property of producing different results from nearly the same starting conditions, similarly may bring two very distant points to the same attractor. This means that the same results may be sometimes achieved using various search methods. An "intelligent enough" agent would probably be able to choose theoretically the best method. However, a SA should probably also look at how robust a given method is, considering previous experience. In this context, **a SA can be defined as an IA capable of avoiding undesired attractors in a representation space, or, if necessary, of using various search methods to reach a given attractor.**

3. Conclusions

The seven approaches to the definition of a SA proposed in Section 2, are summarized in Table 1. This result is only preliminary, and much more work is recommended to improve our present, still incomplete understanding of the concept of a SA.

The behavior of IAs results from the analysis of a given situation and from using the acquired knowledge to select the most promising strategy. It seems that SAs would use some counter-intuitive steps, not necessarily justified by local observations and resulting from accumulated knowledge. For example, SAs would suggest some seemingly illogical steps, predicting that knowledge of a certain kind may be acquired in the future, even if there are no obvious signs that this may happen. Following "intuition," a SA can follow locally inappropriate steps, which may, however, prove beneficial later.

There is a saying that (supposedly by Jan L.A. van Snepscheut [8]) "in theory there is no difference between theory and practice, but in practice there is." When IAs are considered, a natural assumption is that an IA should theoretically always find the best solution. However, if an IA is applied to a real world problem, it may happen that under specific circumstances, not predicted or incorporated in the models, it will fail because of its relative brittleness. A SA is supposed to be much more flexible, mostly because it uses experience (which may not be necessarily fully understood), and some built-in general rules about adaptation and about acquiring new data.

There has been immense progress in applied computer science over the last several decades. Unfortunately, most computer programs still must be run under strict supervision of humans, who must assure that all assumptions are satisfied and that the selected problem solving method is appropriate in a given case. In other words, it is still the experience of humans and their wisdom that makes it possible to use proper automated procedures. We are very curious how much of this wisdom and experience could be transferred to agents to create SAs which could be entirely trusted to operate a nuclear power plant or to design planes. In any case, the authors believe that their paper will initiate a good discussion about our understanding of SAs and about their unique features.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support for their research from the NASA Langley Research Center under the grant NAG-1-01030 and from George Mason University that provided the Graduate Interdisciplinary Research Scholarship for the first author during the academic years 2002-2004.

REFERENCES

[1] Arciszewski, T., Michalski, R.S., Wnek, J., "Constructive Induction: the Key to Design Creativity", *Preprints "Computational Models of Creative Design",* Third International Round-Table Conference, pp.397-426, Heron Island, Australia, December 1995

[2] Arciszewski, T., Sauer, T., Schum, D., "Conceptual Design: Chaos-Based Approach", *Journal of Intelligent and Fuzzy Systems,* in print

[3] Baeck, T., Fogel, D.B., Michalewicz, Z., "Evolutionary Computation: Advanced Algorithms and Operators", Institute of Physics, London, 2000

[4] Campbell, M., "Knowledge discovery in deep blue", *Communications of the ACM*, vol. 42, no. 11, 1999

[5] De Jong, K., "Evolutionary Computation: A Unified Approach", MIT Press, in print

[6] Gero, J., "Computational Models of Innovative and Creative Design Processes", in *Technological Forecasting & Social Change*, Special Issue: *Innovation: The Key to Progress in Technology and Society*, editor Arciszewski, T., vol. 64, numbers 2 & 3, 2000.

[7] Gero J. S., Brazier F. M. T., eds., "Agents in Design, Preprints of the First International Workshop on Agents in Design", University of Sydney, 2002

[8] Giga Quotes, http://www.giga-usa.com/gigaweb1/quotes2/qutoptheoriesx001.htm, accessed 25 June 2003

[9] Kennedy, J., Eberhart, R. C., "Swarm Intelligence", Morgan Kaufmann Publishers, 2001

[10] Michalski, R.S., "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," in *Machine Learning: A Multistrategy Approach, Volume IV*, Morgan Kaufmann Publishers, 1994.

[11] Richards, N., Moriarty, D. E., Miikkulainen, R., "Evolving neural networks to play Go", *Applied Intelligence*, vol. 8, no. 1, pp. 85--96, 1998.

[12] Russell, S. J., and Norvig, P., "Artificial Intelligence: A Modern Approach", Prentice Hall, New Jersey, 1995

[13] Skolicki, Z., Arciszewski, T., "Intelligent Agents in Design", *Proceedings of ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, in print

[14] Sowa, J.F., "Knowledge Representation: Logical, Philosophical and Computations Foundations", Brooks/Cole, 2000

[15] Tecuci, G., Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool, and Case Studies, Academic Press, 1998