On a Formal Framework Relating Knowledge to Information Dependencies

Dagmar Monett*

Humboldt University of Berlin
Department of Computer Science
Artificial Intelligence
Unter den Linden 6
10099 Berlin, Germany
diaz@informatik.hu-berlin.de

Abstract. It is the purpose of this paper to present a formalism dealing with both knowledge of agents and their information dependencies in order to make automation of evolutionary algorithms possible. In this occasion, knowledge of agents is analyzed by defining their dependencies originated by the lack of information, by considering their knowledge about certain data (concepts and knowledge categories), as well as their knowledge about other agents on a multi-agent domain (e.g. agents who handle genetic algorithms). I put emphasis on the relation knowledge-dependency so as to explain both concepts and further actions of agents derived from it. As particular cases of dependency relations, agent and data dependencies of an agent with respect to other agents are here discussed. As application scenario a real-world problem was selected.

1 Introduction

This work is part of an ongoing dissertation research covering the use of intelligent agents to abbreviate the handling of evolutionary algorithms. It intends to combine Artificial Intelligence (AI) techniques with the Evolutionary Computation field. By using intelligent agents, time and resources required to perform the executions of evolutionary algorithms, e.g. Genetic Algorithms (GAs), could be surely reduced. Our research aims to develop a flexible and easy to use Multi-Agent System (MAS) for these purposes.

As application domain, the function optimization of chemical processes by means of GAs was selected, so as to decide whether different mathematical models fit experimental data for a kinetic behavior (see [11] for more). For each problem the user has (i.e. a chemical process or kinetic behavior to be studied) an evolutionary technique (say a GA) should be executed several times in order to achieve the best results. Occasionally, this means running the algorithm a fixed number of times, setting up the number of generations and/or genetic operators and parameter values, and reporting an average result. In general, testing and creating modifications to

* The author makes her PhD at the time of the conference

enhance technique performance are in addition time consuming. Therefore, improvements in the agent-area could be very helpful.

This work presents a framework for dependencies relations among autonomous agents by considering their knowledge about certain information (*concepts* and *knowledge categories*, as described in [12]), as well as knowledge about other agents on a multi-agent scenario. With this purpose, *information dependencies* are defined by relating them to "knowing desired data". I will put emphasis on the relation knowledge-dependency so as to explain both concepts and further actions of agents derived from it.

The remain of the article is conceived as follows: Section 2 contains a brief discussion about the importance of knowledge as a basis for action, and how information dependencies arise when agents lack of certain knowledge. As particular cases of knowledge, details concerning knowledge of agents about concepts and knowledge categories, as well as their relation to information dependencies are presented in Section 3. Knowledge of agents about other agents as a result of the information dependencies they have is also introduced. Section 4 covers some theoretical ideas concerning dependency relations. Finally, Section 5 concludes the paper.

2 KNOWLEDGE OF AGENTS AND THEIR DEPENDENCIES

The notion of dependency has become with time more and more discussed in several areas such as computer science and organizational research. The word *dependency* by itself suggests the necessity of somebody/something by someone/something for a particular purpose or with regularity (e.g. patients need physicians to become healthy; fire needs oxygen to exist). In other words, someone/something lacks of all it needs in order to "correctly" functioning. Inside the scope of AI *dependencies* reflect, for example, what an agent needs from the world (e.g. resources) or other agents for behaving [2]. The words "dependency" and "dependence" are indistinctly used in AI. I will use the first term in this article.

Several authors have already classified from different perspectives what dependencies are and when they arise. In [4], Crowston considers dependencies as resulting from possible combinations of activities using resources. He classifies dependencies well according to how they arise between tasks and resources in three classes:

- *Task-resource dependencies*: One task uses/produces a resource (i.e. everything used or affected by activities).
- *Dependencies between multiple tasks and resources*: Any of the following situations is present: (i) two tasks use/produce a common resource; or (ii) a resource is the effect of one task and a precondition of another (also defined as *producer/consumer dependency*); or (iii) a task is dependent on multiple resources (e.g. using/producing two resources, or using one resource and producing another).

- *Dependencies between tasks or between resources*: tasks or resources are dependent between each other (e.g. two or more parts of a computer system are connected together).

Because different types of dependencies might cause further problems, Crowston as a response to those problems presents coordination. He also describes alternative coordination mechanisms for each type of dependency and considers information as a special kind of resource used or affected by activities. Coordination is defined as a process of managing dependencies between activities, for instance, when the following circumstances arise: shared resources, producer/consumer relationships, simultaneity constraints, and task/subtasks dependencies, as presented in [7]. However, coordination among agents might be problematic in real-world situations: agents must complete their knowledge about the world and about other agents by using all the communication and computation facilities they can, which are not always available or out of conflicts [1], [14].

Other AI approaches present *social* as well as *non social dependencies* implied by plan relations: agents establish relations based on (individual or common) plans to assure a certain level of effectiveness on their actions. For example, in [3], dependence relations set up a social network among agents distinguishing when goals can or cannot be achieved by them. In this way, dependency is defined as the basis for social interaction, and one agent can also devise actions so as to influence other agents that are able to do what it needs. In addition, specifications about data structures on which agents store information about other agents in order to construct their dependence networks are as well defined as *external descriptions* in [15].

Analogous studies concerning degrees of social dependency of agents with respect to a plan or other agents, as well as coordination mechanisms that come out from the existing relationships have been also considered by Ossowski in [14]. MAS and architectures regarding these themes have been developed, too (see [7], [8], [9] for more). Most attempts only describe dependencies in general terms. However, their useful theories have been tried to contribute to equivalent principles. In other words, formalizations with respect to dependency analysis give details about global definitions or domain dependent applications, but only few works have been dedicated to clarify particularities inherent to the type of dependencies itself. For instance, *information dependencies* have been already analyzed in [2], [6] and [7] but not yet complete studied, as far as I know. This paper presents them from a particular perspective.

2.1 Knowledge as the Basis for Action

Acquiring and handling information are crucial phases in every autonomous system. This means, the more informed an agent is about the world and itself, the better it can act or autonomously behave. Castelfranchi defines autonomy in [2] as “the complement of dependency”, and relates several dimensions of autonomy/dependency situations where agents are involved in. One of the above dimensions, the *information autonomy/dependency*, reflects information that an

agent does not directly access to, while other agent has this information and provides it. In this case, the former agent is defined as dependent on the latter *as for information*.

In our context, we consider non-possessed information as a special kind of knowledge that an agent lacks of. If this information is essential for further agents' actions, then it is important to define the ways the agents have to acquire, process, or simply know it, so as to behave as an integrated part of the system. As a result, coordination should arise as a phase where information is obtained if an agent depends on others as for information (i.e. the information is possessed by other agents or by the user). In other words, the agent has an *information dependency* derived from its lack of certain knowledge and should then coordinate its actions with others in order to acquire the information it needs for a future behavior.

Several notational formalisms have being developed to represent knowledge (e.g. First Order Logic and production rules) but no one is as expressive as the natural language and the human reasoning are. Knowledge could be explicitly declared but it can also be deduced from other knowledge. Implicit reasoning by rules, for instance, describes modularly knowledge and follows a common sense as would be pursued by a human expert. Since knowledge must and needs to be programmed, expertise or professional knowledge to solve a problem is indispensable when designing a system. Therefore, the expressiveness and inferential capabilities of a system depend on how the knowledge is first represented and then processed.

Most of the times incompleteness of knowledge, vagueness and imprecision derived from characteristics of real-world problems should be considered. Consequently, making decisions and assumptions regarding unknown information is not an expeditious theoretical or practical task. Some difficulties might also arise by considering real-world situations: real problems are complex and cannot always be symbolically represented. Besides, a knowledge base (when used) could be not extensive enough (regarding expressiveness and not properly the size) so as to cover all possible cases. Also the rules for acquiring, completing or updating knowledge could be not as expressive as needed. Moreover, a new problem does not have always to have a similar in the past, the information can be incomplete or vague due to practical reasons, and measurement errors could be present regarding the equipment precision with which problem data are obtained. As a consequence, either explicit or implicit knowledge from agents tries to include all possible situations so as to determine their later behaviors, however a total completeness of the knowledge cannot be assured. Thus, agents will necessarily have certain dependencies on others for behaving.

Basically, intelligent systems base their actions on the knowledge representation and processing capabilities they have implemented. Intelligent agents, for example, determine its behavior by its own experience, and this experience is certainly knowledge. Then, knowledge is an important basis for action. For instance, in [13] the authors focus on:

"The relation between knowledge and action is a fundamental one: a processor in a computer network (or a robot or a person [...]) should base its actions on the knowledge (or information) it has."[1]

The forthcoming sections describe formal aspects for a model of knowledge and information dependencies, support for the agents' actions on the MAS I develop.

3 AN APPROACH TO REPRESENT KNOWLEDGE

In this section some MAS features concerning agents knowledge are faced. I do not provide at this time details concerning neither the system architecture nor its functioning. I merely describe its general purpose in order to make further definitions and examples more understandable by the reader.

In the sequel, agents will be represented in a finite set A of agents. For simplicity and maintaining the notations used in [10] I will name them *specialist* agents, *operator* agents, and *result* agents according to their functionality on the MAS. For example, *operator* agents are used to execute an evolutionary algorithm (e.g. a GA) several times and with different genetic parameters so as to handle it just as an expert would do. This means the user does not have to know the method particularities in order to use it: most activities are delegated to intelligent agents. Next section synthesizes various formal concepts related to kinds of knowledge, which are considered as a start point for further sections in this work.

3.1 Definitions Concerning Knowledge

As mentioned earlier, I will make use in this paper of some definitions from previous work. They are synthesized as follows:

A *knowledge category* $C^i \in C$ (C stands for the set of knowledge categories) refers a particular set of information that is required by the agents from A. For example, the knowledge category $C^{SOLMETHOD}$ contains parameters and requirements (e.g. population size, values for the genetic operators, etc.) related to a solution method (e.g. a GA).

A *concept* $c \in C^i$ is a pair (*attribute*, *value*) such that $C^i = \{c_j=(p_j, v_j) \mid p_j$ is an attribute in C^i and v_j corresponds to its value$\}$. For example, a value of 100 individuals conforming a population in a GA corresponds to the concept *population size* from the $C^{SOLMETHOD}$ knowledge category.

An agent $a \in A$ *does not know* a concept c from a knowledge category $C^i \in C$ if $\varphi(a, c)$=**no** (with $\varphi: AxC \rightarrow \{$**yes**, **no**$\}$). That is, agent a does not know the attribute value

[1] Moses, Dolev and Halpern discuss in their paper the relationship between knowledge, action and communication in a distributed environment by looking at a number of variants of the *cheating husbands* puzzle. Also in [5] the authors present a formal model capturing interactions between knowledge and action in distributed systems.

on c, denoted by $-a_\varphi c$. On the contrary, if $\varphi(a, c)$=**yes** then the agent *a knows* the concept c from the knowledge category C^i, denoted by $a_\varphi c$. For example, an *operator* agent directly related to the execution of a GA needs to know the *population size* that a *specialist* agent obtains from the user needs.

Let $a \in A$ be an agent. The agent' *knowledge about a knowledge category* $C^i \in C$ is defined as follows:

a) An agent *a does not know* or *ignores* a knowledge category C^i, denoted by $-a_\varphi C^i$ (read "agent *a* does-not-know/ignores C^i"), if agent *a* does not know its concepts, i.e. agent *a* has no information about the attribute values the knowledge category has. This means, $\varphi(a, c_m)$=**no** (or $-a_\varphi c_m$) $\forall\ c_m \in C^i$, with m in $\{1, ..., D^i\}$. For example, agents who handle a GA (i.e. *operator* agents) cannot execute it before the user have defined the problem data.

b) An agent *a* has *partial knowledge* about a knowledge category C^i or it knows C^i *partially*, denoted by $\sim a_\varphi C^i$ (read "agent *a* knows C^i partially"), if agent *a* knows at least one concept from C^i and, at the same time, ignores at least another concept from C^i. This means, $\exists\ c_m \in C^i$ such that $\varphi(a, c_m)$=**yes** (or $a_\varphi c_m$) with m in $\{1, ..., D^i\}$, and simultaneously $\exists\ c_n \in C^i$ such that $\varphi(a, c_n)$=**no** (or $-a_\varphi c_n$) with n in $\{1, ..., D^i\}$.

c) An agent *a knows* a knowledge category C^i or it knows C^i *completely*, denoted by $a_\varphi C^i$ (read "agent *a* knows C^i completely"), if it knows all concepts the knowledge category has. This means, $\varphi(a, c_m)$=**yes** (or $a_\varphi c_m$) $\forall\ c_m \in C^i$, with m=1, ..., D^i. Following the example in a), agents can operate a GA if they have all the parameters and data the GA needs.

It has been assumed that the dimension of a knowledge category C^i is finite and it is given by D^i, i.e. the number of concepts the knowledge category has.

I will now relate the above definitions to the concept of dependency.

3.2 Relating Knowledge to Dependencies

As discussed in Section 2, agents might have information dependencies derived from their lack of certain knowledge. From the viewpoint of concepts, this means that an agent $a \in A$ does not know at least a concept c_m from a knowledge category C^i or $-a_\varphi c_m$ (i.e., $\varphi(a, c_m)$=**no**). In other words, if agent *a* needs the concept c_m for further actions (observe that do not knowing a concept does not always implies being impossible to act) then agent *a* should "ask" other agents (or the user) to. This means, agent *a* has an *information dependency with respect to a concept* (i.e. c_m). For example, a concept related to stop criteria for a GA is a essential one. An *operator* agent must know this concept (i.e. its value) in order to appropriately execute the GA.

These considerations can be extended to the case of knowledge categories because concepts conform them: if agent *a* partially knows the knowledge category C^i (resp. ignores it), then agent *a* has an *information dependency with respect to a knowledge category* (i.e. C^i). This means, $\exists$ (resp. $\forall$) $c_n \in C^i$ such that $\varphi(a, c_n)$=**no** (or

$-a_{\varphi}c_n$), with n in {1, ..., D^i}. In the same way, if agent a knows the knowledge category C^i it means that it will not have information dependencies related to C^i, that is the same as being *autonome* with respect to the information contained on C^i (by following similar definitions from [2]).

But this does not only solve the dilemma: an agent needs to know if other agents (or the user) have the information it needs in order to ask for and then use this information. This problematic is focused on in the following section.

3.3 Knowledge About Other Agents

The knowledge an agent has is not limited to its knowledge about concepts and knowledge categories, but the knowledge about other agents, about itself, and so on. For this reason, different *levels* of knowledge are here considered so as to facilitate agents' interactions on a social environment. In this section I will define other kinds of them.

When communication is not guaranteed it is impossible to establish both a robust model of knowledge and message passing. For instance, recalling *specialist* and *operator* agents from our MAS, the former ones receive data from the user and send them to the latter ones. A *specialist* agent must know if the *operator* agents it communicated with got everything it sent to. Hence *operator* agents must know what should be sent, who should send it, as well as whom ask for incomplete information before planning and accomplishing further actions. All agents here need to have information about the others.

Definition 1. An agent $a_1 \in A$ knows another agent $a_2 \in A$ is denoted by $a_1 \varphi a_2$. If agent a_1 does not know agent a_2, then the notation is $-a_1 \varphi a_2$.

This indicates for us, that agent a_1 has its "connections" with agent a_2 already defined (or the information about agent a_2 in its external descriptor as in [15] already completed). Then, if the communication channel has been previously activated, agent a_1 can send agent a_2 information (say data). But the contrary can also happen: that agent a_2 does not already know agent a_1. For a successful communication between the agents, both directions should be well established.

In terms of information, it is important that an agent expecting some data from other agent could verify if the other one has the information it needs in order to ask for this information:

Definition 2. An agent $a_1 \in A$ knows that an agent $a_2 \in A$ knows a knowledge category $C^i \in C$ is denoted by $a_1 \varphi (a_2 \varphi C^i)$.

This means $a_2 \varphi C^i$ holds but probably either $-a_1 \varphi C^i$ or $\sim a_1 \varphi C^i$ holds (see definitions from previous section). This signifies that agent a_1 needs agent a_2 in order to complete its information, i.e. satisfy $a_1 \varphi C^i$ because it does not know all concepts on C^i. In other words, agent a_1 has an *information dependency regarding agent* a_2 with respect to a knowledge category (i.e. C^i). For instance, when a GA execution concludes *result* agents depend on *operator* agents in order to compact obtained results and present them to the user.

In this situation it is supposed that agent a_2 completely knows C^i. Considering partial and no knowledge about knowledge categories we can formulate similar definitions. In essence, agent a_2 should at least know the concepts agent a_1 needs: both $a_2 \varphi c_m$ and $-a_1 \varphi c_m$ hold with m in {1, ..., D^j} and $c_m \in C^i$. These relations are very useful in mechanisms retrieving needed information, or just to verify if it comes from the appropriate source.

Other properties can be also described. For example, the following holds: if $a_1 \varphi (a_2 \varphi C^i)$ then $a_1 \varphi a_2$, where a_1 and a_2 are two agents on a finite group A of agents and $C^i \in C$ is a knowledge category. However, if exists a knowledge category $C^i \in C$ such that agent a_1 needs to know some of its concepts (either $-a_1 \varphi C^i$ or $\sim a_1 \varphi C^i$ holds), and simultaneously $a_2 \varphi C^i$ holds, then agent a_1 can assume that it knows that agent a_2 knows the knowledge category C^i because $a_1 \varphi a_2$ holds. This property is only an example of a simple one who arises in such a system.

These observations lead me to consider dependency relations that are analyzed in the following section.

4 DEPENDENCY RELATIONS

Let me present once again an example: if there are defined several *operator* agents solving the same problem by using different evolutionary algorithms (e.g. GAs, Evolutionary Strategies, etc.), all of them will need the same information about the real problem itself being solved, or about the *specialist* and *result* agents to communicate with, to name a few. Their knowledge about other agents will determine their behaviors because they depend on others to get the information they need. In other words, the behavior of an agent is characterized by the dependencies the agent has. As we discussed earlier, dependencies allow an agent distinguish what should it know in order to complete its knowledge for further actions. Formally, we define these ideas considering two types of dependency relations, *data* and *agent dependencies*, as follows:

Definition 3. An agent $a_1 \in A$ has a *data dependency* on another agent $a_2 \in A$ with respect to a knowledge category $C^i \in C$, denoted by *D-dep*(a_1, a_2, C^i), if $a_1 \varphi a_2$ and $a_2 \varphi C^i$ hold, $a_1 \varphi (a_2 \varphi C^i)$ holds, but either $-a_1 \varphi C^i$ or $\sim a_1 \varphi C^i$ holds, too.

Similar definitions can be also considered by assuming partial and no knowledge of agent a_2 with respect to C^i (i.e. $-a_2 \varphi C^i$ or $\sim a_2 \varphi C^i$). With all of them, agents being dependent on others as for information, i.e. concepts from knowledge categories, as we mentioned in Section 3.3 are finally formalized.

Evidently, if an agent depends on another one with respect to a knowledge category, then it has a dependency with respect to the other agent:

Definition 4. An agent $a_1 \in A$ has an *agent dependency* regarding another agent $a_2 \in A$ with respect to a knowledge category $C^i \in C$, denoted by *A-dep*(a_1, a_2), if *D-dep*(a_1, a_2, C^i) holds.

With the above definition the dependency on other agents is resumed. It is reasonable to think the fact that if agent a_1 does not know agent a_2 (or $-a_1 \varphi a_2$ by Def.

1), then it cannot have an agent dependency from agent a_2 since agent a_2 does not exist for agent a_1. However, agent a_1 knows that it needs some information that another agent possibly has, and then it will potentially have an agent dependency in the future. By designing such agents it is practicable to define a *set of dependencies* (i.e. for each agent on A) such that agents are able to know, when they are created, all possible data and agent dependencies they will potentially have. This set of dependencies is conceived as a *dependence network* in [15]: agents construct such a network to represent action/resource dependencies regarding the others based on dependency relations they have.

So far there have been defined some features concerning knowledge of agents and their dependencies. Information about GAs parameters and dependencies of agents related to them were cited as examples. Further work will focus on other aspects related to coordination mechanisms: for example, how the coordination complexity depends on the dependency complexity and how it can be applied.

5 CONCLUSIONS

This work presented part of a taxonomy including knowledge of agents in several situations by considering their information dependencies on a multi-agent domain. Why knowledge of agents about concepts and knowledge categories constitutes a helpful support in defining and understanding information dependencies was analyzed. Hence, in order to formulate an ontology relating knowledge to information dependencies some of these aspects were formalized.

The contribution of this work is mainly theoretic. It covers the study of information dependencies in the context of a specific MAS but it can be simply extended to other social scenarios: basic definitions would remain the same. Up to now, we use these concepts on an ongoing PhD research so as to partially or completely automate evolutionary techniques applied to inverse problems on Chemical Kinetics.

I believe this approach is useful in order to formalize AI techniques dealing with evolutionary algorithms. The latter expands their boundaries establishing a necessary contact with the former ones.

REFERENCES

1. Barbuceanu, M., and Fox, M. S.: The Architecture of an Agent Building Shell. In M. Wooldridge, J. P. Mueller and M. Tambe (eds.), Intelligent Agents II: Agent Theories, Architectures and Languages, Springer Verlag, *Lecture Notes in Artificial Intelligence*, 1037 (1995).
2. Castelfranchi, C.: Founding Agent's 'Autonomy' on Dependence Theory. In W. Horn (Ed.), Proceedings of ECAI'2000, Berlin, Germany, IOS Press (2000) 353–357.

3. Castelfranchi, C., Miceli, M., and Cesta, A.: Dependence Relations among autonomous agents. *Decentralized A.I.*-3, E. Werner and Y. Demazeau (eds.), Elsevier Science Publishers B. V. (1992) 215–227.
4. Crowston, K.: A Taxonomy of Organizational Dependencies and Coordination Mechanisms. Technical Report #174, Cambridge, MA: MIT Centre for Coordination Science (1994).
(at http://ccs.mit.edu/papers/CCSWP174.html).
5. Halpern, J. Y., and Fagin, R.: Modelling knowledge and action in distributed systems, *Distributed Computing*, Sringer Verlag, 3 (1989) 159–177.
6. Horling, B. C.: A Reusable Component Architecture for Agent Construction. UMass Computer Science Technical Report #1998-49, University of Massachusetts, October (1998).
(at http:// mas.cs.umass.edu/~bhorling/papers/1998-49/).
7. Malone, T. W., and Crowston, K.: The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, *26* (1) (1994) 87–119.
8. Malyankar, R. M.: A Pattern Template for Intelligent Agent Systems. Workshop on Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, Agents'99, Seattle, WA (1999).
9. McCabe, F. G., and Dale, J.: Agent Configuration Management. Fujitsu Laboratories of America, Technical Memorandum FLA-NARTM99-04, Fujitsu Laboratories of America (1999).
10. Monett, D.: Towards an Intelligent Agent Design for Handling Genetic Algorithms. In H.-D. Burkhard, L. Czaja, A. Skowron and P. Starke (eds.), Proceedings of CS&P 2000, Informatik-Bericht Nr. 140, Humboldt University Berlin, Vol. 1 (2000) 149–155.
11. Monett, D.: On the automation of evolutionary techniques and their application to inverse problems from Chemical Kinetics, In Proceedings of the GECCO'01 Graduate Student Workshop, San Francisco, California, USA (2001) 429–432.
12. Monett, D.: On Modelling Knowledge in a Multiagent Scenario: A Preliminary Report. In L. Czaja (Ed.), Proceedings of CS&P 2001, (Zakład Graficzny UW, zam. 583/2001) Warsaw, Poland (2001) 162–168.
13. Moses, Y., Dolev, D., and Halpern, J.Y.: Cheating husbands and other stories: A case study of knowledge, action, and communication, *Distributed Computing*, Springer Verlag, 1 (1986) 167–176.
14. Ossowski, S.: Co-ordination in Artificial Agent Societies: Social Structure and its Implications for Autonomous Problem-Solving Agents. In J. G. Carbonell and J. Siekmann (eds.), *Lecture Notes in Artificial Intelligence* (Subseries LNAI), 1535, Springer Verlag (1999) (entire issue).
15. Sichman, J. S., Conte, C., Castelfranchi, C., and Demazeau, Y.: A Social Reasoning Mechanism Based on Dependence Networks. In A. Cohn (Ed.), Proceedings of ECAI'94, Amsterdam, The Netherlands, John Wiley & Sons (1994) 188–192.