Anthropopathy and its Assessment in Virtual Entities

Cesar Analide, Paulo Novais, José Machado & José Neves

Department of Informatics
University of Minho
4710-057 Braga, PORTUGAL

Phone: +351 253 604 470
Fax: +351 253 604 471
{analide, pjon, jmac, jneves}@di.uminho.pt

Abstract. The introduction and the natural evaluation of virtual entities presenting human like feelings and behaviours, living in virtual worlds, being based on agents, organizations or other kind of artefacts, has been made, almost exclusively, by an evaluation of such characteristics and assumptions, in terms of a set of quantitative variables. In this paper, it is presented an alternative way to analyse and evaluate an intelligent's system body of knowledge in terms of its anthropopathic potential, that considers quantitative, qualitative and incomplete information, through and extension to the language of logic programming.

1 INTRODUCTION

There has been experiences in the fields of Computer Science, Artificial Intelligence (AI) and MultiAgent Systems (MAS) that foresees an approximation of these disciplines and those of Social Sciences, namely in the areas of Anthropology, Sociology, and Psychology.

Much work has been done in terms of the humanization of the behaviour of virtual entities, by expressing human like feelings and emotions; work presented in [11] [12] detail studies and propose lines of action that consider the way to assign emotions to machines. Attitudes like cooperation, competition and socialization of agents [5] are explored, for example, in the areas of Economy [4] and Physics [9], as it is the case of the "*El Farol Bar Problem*", the "*Minority Game*" and the "*Iterated Prisoner's Dilemma*". In [6] and [7] is recognized the importance of modeling the virtual agent's mental states in a human like form.

Indeed, an important motivation to the development of this project comes from the authors work that has been done on the intersection of the disciplines of AI and The Law, that enforced new forms of knowledge representation and reasoning in terms of an extension to the language of logic programming (i.e., the Extended Logic Programming (ELP) [1] [11] [13]). On the other hand the use of null values and the enforcement of exceptions to characterize the behaviour of intelligent systems is in itself another justification for the adoption of these formalisms in this knowledge arena.

Knowledge representation, as a way to describe the real world, based on mechanical, logical or other means, will be, always, a function of the systems ability to describe the existent knowledge and their associated reasoning mechanisms. Indeed, in the conception of a knowledge representation system, it must be taken into attention:

Existent Information – it will not be known in all its extension.

Observed Information – that is acquired by the experience; it must be taken into account that the observed information depends on the observer, in the same way a measurement is influenced by the instrument that measures it. Indeed, a verbal statement of an event depends on the observer education, on his state of mind, his prejudices (only to state a few).

Represented Information – with respect to a certain objective, it may be (ir)relevant to represent a given set of information. In spite of all exceptions, it is possible that observations made by different individuals, with distinct education and motivations, may show the same set of fundamental data, function of the utility of the information obtained. This is the information that must be represented and understood.

In a classical logical theory, the proof of a question is made in terms of being true or false, or in terms of representing something about which one could not be conclusive. In spite of that, in a logic program the answers to questions are only of two types: they are true or false. This is due to the fact that a logic program shows some limitations in terms of knowledge representation (it is not allowed explicit representation of negative information); in addition, in terms of an operational semantics, it is applied the Closed World Assumption (CWA) to all the predicates.

The generality of the programs written in logic represents implicitly negative information, assuming the application of reasoning according to the CWA. An extension of a logic program may comprise negative information [1] [11], as well as directly describe the CWA for some predicates. Consequently, it is possible to distinguish three types of conclusions for a question: ***true***, ***false*** or, when there is no information allowing inferring one or another, the answer will be ***unknown***.

2 PRELIMINARIES

This work is supported by the developments in [2] where the representation of incomplete information and the reasoning based on partial assumptions is studied, using the representation of null values [3] [10] to characterize abnormal or exceptional situations.

2.1 Null Values

The identification of null values emerges as a strategy for the enumeration of cases, for which one intends to distinguish between situations where the answers are ***known*** (true or false) or ***unknown*** [3] [13].

The representation of null values will be scoped by the ELP. In this work, it will be considered two types of null values: the first will allow the representation of unknown values, not necessarily from a given set of values, and the second will represent unknown values, from a given set of possible values.

Consider the following as a case study to show some examples of how null values can be used to represent unknown situations. Consider the implementation of a time-table to express the departure of trains, through the predicate:

```
connect: City × Time
```

where the first argument denotes the city of departure and the second represents the time of arrival (e.g., `connect( guimarães,17:00 )` denotes that the Guimarães' coming train is expect to arrive at 17 o'clock, Program 1).

```
connect( guimarães,17:00 )
¬connect( C,T ) ←
        not connect( C,T )
```

Program 1: Extension of the predicate that describes arrivals at the train station

In Program 1, the symbol ¬ denotes the strong negation, denoting what should be interpreted as ***false***, and the term `not` designates negation by failure.

Unknown

Following the example given by Program 1, one can admit that the connection from Oporto has not yet arrived. This situation will be represented by a null value, of the type ***unknown***, that should allow the conclusion that the connection exists, but to which it is not possible to be affirmative with respect to the arrivals time (Program 2).

```
connect( guimarães,17:00 )
connect( oporto,⊥ )
¬connect( C,T ) ←
        not connect( C,T ) ∧ not exception( connect( C,T ) )
exception( connect( C,T ) ) ←
        connect( C, ⊥ )
```

Program 2: Information about Oporto connection, with an unknown delay

Symbol ⊥ represents a null value of an undefined type, in the sense that it is a representation that assumes that any value is a potential solution but without given the clue to conclude about which value one is speaking about. Computationally, it is not possible to determine, from the positive information, the arrivals time of the Oporto's connection; by the description of the exception situation (fourth clause from Program

2, the closure of predicate connect), it is discarded the possibility to be assumed as false any question on the specific time of arrival of that connection.

Unknown but Enumerated

Consider now the example in which the time of arrival of the Lisbons' connection is foreseen to 18 o'clock, but is 15 minutes delayed. It is not possible to be affirmative regarding the arrival at 18:00 or at 18:01 or even at18:15. However, it is false that the train will arrive at 16:16 or at 17:59. This example suggests that the lack of knowledge may only be associated to an enumerated set of possible values.

```
connect( guimarães,17:00 )
connect( oporto,⊥ )
¬connect( C,T ) ←
          not connect( C,T ) ∧ not exception( connect( C,T ) )
exception( connect( C,T ) ) ←
          connect( C, ⊥ )
exception ( connect( lisbon,T ) ) ←
          T ≥ 18:00 ∧ T ≤ 18:15
```

Program 3: Representation of the connection with a 15 minutes delay

The exception occurs to the time interval 18:00…18:15. It is unknown that the Lisbon's connection will arrive at 18:05 or at 18:10; it is false that it will arrive at 17:55 or at 18:20.

2.2 Interpretation of Null Values

To reason about the body of knowledge presented in a particular knowledge, set on the base of the formalism referred to above, let us consider a procedure given in terms of the extension of a predicate called demo, using ELP as the logic programming language. Given a question it returns a solution based on a set of assumptions. This meta-predicate will be defined as:

```
demo: Question × Answer
```

where Question denotes a theorem to be proved and Answer denotes a truth value: True (*T*), False (*F*) or Unknown (*U*) (Program 4).

```
demo( Q, T ) ← Q
demo( Q, F ) ← ¬Q
demo( Q, U ) ← not Q ∧ not ¬Q
```

Program 4: Extension of meta-predicate demo

The first clause of Program 4 sets that a question it is to be answered with appeal to the knowledge base positive information; the second clause denotes that the question is proved to be false with appeal to the negative information presented at the knowledge base level; the third clause stands for itself.

3 RECOGNITION OF ANTHROPOPATHIC QUALITIES

Based on the assumptions presented before, it is possible to establish mechanisms to analyze and process the information available in a way that turns feasible the study of the behaviour of virtual entities, in terms of its personification. Situations involving forgetfulness, remembrance, learning or trust can be analyzed in the way proposed in this work; i.e., the description of abnormal situations, declared as exceptions to a predicate extension, made possible one's goals.

3.1 Characterization of a Problem

Consider the following example, built up to illustrate the practical application of what is the main contribution of this work.

```
parent( carlos,joão )
¬parent( P,S ) ←
          not parent( P,S ) ∧ not exception( parent ( P,S ) )
```

Program 5: Excerpt of an extended logic program, representing knowledge at a time t_i

In Program 5 there is an axiom stating that Carlos is a parent of João. Assuming that this is all the knowledge available at instant t_i, the second clause of Program 5 enforces that it must be considered false all other situations where there is a lack of information and that are not being treated as exceptions.

Suppose that, an instant later, t_j, the knowledge evolves in such a way that it may be represented as shown in Program 6.

```
¬parent( P,S ) ←
          not parent( P,S ) ∧ not exception( parent ( P,S ) )
exception( parent( carlos,joão ) )
exception( parent( luís,joão ) )
exception( parent( pedro,joão ) )
```

Program 6: Knowledge base excerpt, at instant t_j

At a third instant of time, t_k, the knowledge base is shown as Program 7.

```
parent( ⊥,joão )
¬parent( P,S ) ←
          not parent( P,S ) ∧ not exception( parent(P,S ) )
exception( parent( P,S ) ) ←
          parent( ⊥,S )
```

Program 7: Excerpt of the program that shows how the knowledge base evolves, between instants t_j and t_k

Looking to the way the knowledge base evolved, between instants t_j to t_k, one may say that the information has been loosing specificity. In the beginning it was known that Carlos was a parent of João (t_i); after that, it was only known that the parent of João was Carlos, Luís or Pedro (t_j); finally, in a third instant, the system only knows that João has a parent, but cannot be conclusive about who is, in fact, such a person; it is not also possible to state that João has not a father.

Consequently, in terms of the temporal axis $t_i \rightarrow t_j \rightarrow t_k$, one may say that the knowledge evolution has taken a form of forgetfulness, leading to the emptying of the knowledge base knowledge. However, taking the knowledge evolution in the other way around; i.e., $t_k \rightarrow t_j \rightarrow t_i$, a similar analysis leads to the conclusion that the knowledge base learned something, showing that the knowledge base evolves in a way that secures its information.

3.2 The System Semantics

Last but not least, it is now possible to pay some attention to the humans like attributes to be represented at a system level, considering the ELP as the language to describe its knowledge base(s) or theory(ies). Consequently, the objective here is to define those mechanisms that will allow the advent of computational agents, at the system level, with humans' like properties and behaviours, making the way to a certain kind of personification of those computational entities.

Let us consider the Program 5, referred to above, that describes the state of the system at instant t_i, where is questioned who is João's parent. In terms of the demo meta-predicate, one may have:

(i) $\forall_{(P)}$: demo(parent(P,joão),*T*)?
$\angle$ successful
$\forall_{(P)}$: demo(parent(P,joão),*F*)?
$\angle$ unsuccessful
$\forall_{(P)}$: demo(parent(P,joão),*U*)?
$\angle$ unsuccessful

This question is answered in terms of the knowledge base positive information that states that Carlos is João's parent. It is now possible to determine the amount and quality of the information that was used in this round. In other words, one intends to find the set of all the solutions that could contribute to solve the question referred to above, namely:

(ii) $\forall_{(P,S)}$: findall(P,demo(parent(P,joão),*T*),S)?
$\angle$ S = [carlos]

Let us now consider the Program 6, referred to above, and in this context, to endorse the same question as in (i). One may have:

(iii) $\forall_{(P)}$: demo(parent(P,joão),*T*)?
$\angle$ unsuccessful
$\forall_{(P)}$: demo(parent(P,joão),*F*)?
$\angle$ unsuccessful

$\forall_{(P)}$: demo(parent(P,joão),U)?
∠ successful

i.e., the question is solved but the answer is vague. This means that endorsing the question as in (ii) it will give rise to an empty set of solutions, when invoked in terms of the meta-predicate demo. One may have:

(iv) $\forall_{(P,S)}$: findall(P,demo(parent(P,joão),U),S)?
∠ S = [ ]

This situation denotes that there are clauses defined as exceptions to the extension of predicate *parent*, allowing the solution to be unknown, U. One may now turn to the exceptions in order to evaluate the answer. One may have:

(v) $\forall_{(P,S)}$: findall(P,exception(parent(P,joão)),S)?
∠ S = [carlos, luís, pedro]
$\forall_{(S,N)}$: length(S,N)?
∠ N = 3

In this case, attending to the fact that there are three exceptions to the predicate extension, the vagueness of the data is set to $^1/_3$.

Finally, let us consider the case describe by the Program 7, referred to above. By the application of the same procedures as in (i), one may have:

(vi) $\forall_{(P)}$: demo(parent(P,joão),T)?
∠ unsuccessful
$\forall_{(P)}$: demo(parent(P,joão),F)?
∠ unsuccessful
$\forall_{(P)}$: demo(parent(P,joão),U)?
∠ successful

i.e., the solution to the question is undefined. In this case, and acting as in (ii), one is presented with a specific result:

(vii) $\forall_{(P,S)}$: findall(P,demo(parent(P,joão),U),S)?
∠ S = [⊥]
$\forall_{(S,N)}$: lenght(S,N)?
∠ N = ∞

i.e., the evaluation of the truth value to assign to the solution falls back upon a mechanism that starts from an unlimited set of possible solutions. It is to be understood that the cardinality of such a set tends to infinite.

4 CONCLUSIONS

ELP proved to be a well adequate tool for knowledge representation and reasoning, in particular when one intend to endorse situations where the information is vague or incomplete, which is the case when there is the intention to represent at the agent's level properties and attitudes only found in the humans. The use of these techniques, in particular in intelligent systems, are adequate to endorse problems where the knowledge of several agents has to be diffused and integrated, and the agent reasons about the knowledge or the behaviour of their peers, in a competitive and/or collaborative way.

Referências

1. Alferes, J., Pereira, L., Przymusinski, T.: Classical Negation in Nonmonotonic Reasoning and Logic Programming. Journal of Automated Reasoning, vol. 20, pp. 107-142 (1998)
2. Analide, C.: Knowledge Representation and Reasoning in Hierarchical Structures. MSc Thesis, Department of Informatics, Universidade do Minho (1996)
3. Analide, C., Neves, J.: Representing Incomplete Knowledge. 1ª CAPSI, October, Universidade do Minho, Guimarães, Portugal (2000)
4. Arthur, W.: Inductive Reasoning and Bounded Rationality (The El Farol Bar Problem). Proceedings of the American Economics Review, 84:406 (1994)
5. Bazzan, A., Bordini, R.: Evolving agents with moral sentiments in an iterated prisoner's dilemma exercise. Proceedings of the 2nd Workshop on Game Theoretic and Decision Theoretic Agents, 4th ICMAS, 10-12 July, Boston, 13-25 (2000)
6. Bazzan, A., Bordini, R., Vicari, R., Wahle, J.: Evolving populations of agents with personalities in the minority game. Proceedings of the International Joint Conference: 7th IBERAMIA and 15th SBIA, 19-22 November, Atibaia-SP, Brazil, number 1952 in Lecture Notes in Artificial Intelligence, 166-175. Berlin: Springer-Verlag (2000)
7. Castelfranchi, C., Rosis, F., Falcone, R.: Social Attitudes and Personalities. In Agents, Socially Intelligent Agents, AAAI Fall Symposium Series 1997, MIT in Cambridge, Massachusetts, November 8-10 (1997)
8. Challet, D., Zhang, Y.: On the Minority Game: Analytical and Numerical Studies, Physica A 256, 514 (1998)
9. Gelfond, M.: Logic Programming and Reasoning with Incomplete Information, in Annals of Mathematics and Artificial Inteligence, 12, pp. 89-116 (1994)
10. Neves, J.: A Logic Interpreter to Handle Time and Negation in Logic Data Bases, in Proceedings of ACM'84, The Fifth Generation Challenge, pp. 50-54 (1984)
11. Ortony, A., Clore, G. L., and Collins, A.: The Cognitive Structure of Emotions. Cambridge, UK: Cambridge University Press (1988)
12. Picard, R.: Affective Computing, MIT Press, Cambridge (1997)
13. Traylor, B., Gelfond, M.: Representing Null Values in Logic Programming. Proceedings of the ILPS'93 Workshop on Logic Programming with Incomplete Information, pp. 35-47, Vancouver, Canadá (1993)