

Artificial intelligence agent for psycho-affective accompaniment^{*}

Gabriel Cervantes¹, Andrés Munguía², Mauricio Osorio³, and Anahí de los Santos Gómez¹

¹ UDEMEX, Toluca México -

{gabriel.cervantes, anahi.delosantos}@udemex.edu.mx

² Grupo de Psiquiatría y Psicología Infantil de Puebla, Puebla, Mexico -

andres.munguiabs@udlap.mx

³ Universidad de las Américas Puebla, Ex Hacienda Sta. Catarina Mártir S/N. San Andrés Cholula, Puebla 72810, MX - mauricioj.osorio@udlap.mx

Abstract. We propose a methodology based on an agent that, using artificial intelligence, creates class notes for a specific course, which are personalized and transformed, through the use of the python language, in conversations. Personalization is made based on a dynamic database that is built with the student. From this, the information obtained allows the elaboration of a profile that includes: learning styles, motivations, interests and emotional situations that limit learning, among other information. In conversations cognitive elements are interpreted through logical formulas. For example, by detecting a user with a profile prone to anxiety and a superficial learning approach, the DLV software traces routes to offer the user a variety of exercises based on mindfulness and other techniques that are related to their learning style, with the intention of helping the user to facilitate their state of receptivity and favor the learning process. The agent receives constant feedback to redefine the characteristics and preferences of the user to be used in their next interaction.

Keywords: Intelligent Agents · Knowledge Representation · Answer Sets Programming · Mindfulness.

^{*} Supported by UDEMEX.

1 Introduction

We propose a methodology based on an agent that, using artificial intelligence, creates class notes for a specific course, which are personalized and transformed, through the use of the python language, in conversations. Our artificial intelligence approach is based on knowledge representation. "Knowledge representation and reasoning is the foundation of artificial intelligence, declarative programming, and the design of knowledge-intensive software systems capable of performing intelligent tasks" [1].

Personalization is made based on a dynamic database that is built with the student. From this, information is obtained that allows the elaboration of a profile that includes: learning styles, motivations, interests and emotional situations that might limit learning, among other information. On this regard, we defined a logical theory that defines our knowledge about learning styles. We name VC our methodology and we will refer to it in this way in the rest of our paper. In sensing, a category well accepted in the context of learning styles we can find that: Sensors often like solving problems by well-established methods and dislike complications and surprises.

We claim that the given sentence has the suitable ingredients to be formalized using answer set programming (ASP). It has already a standard structure of a default logical rule. Note that modeling "oftenly" is considered as a major target in ASP.

In fact it is standard to rewrite the given sentence as: If given sensor S is not an exception then X like solving problems by well-established methods and dislike complications and surprises.

Its abstract logical form could be represented as: $\neg ex(S) \rightarrow (X \wedge Y \wedge Z)$

where

X stand for: Sensors like solving problems by well-established methods.

Y stand for: Sensors dislike complications.

Z stand for: Sensors dislike surprises.

Observe that $\neg ex(S) \rightarrow (X \wedge Y \wedge Z)$ is strongly equivalent (see [2, 3]) to: $(\neg ex(S) \rightarrow X) \wedge (\neg ex(S) \rightarrow Y) \wedge (\neg ex(S) \rightarrow Z)$.

That we can split into 3 formulas:

$$\neg ex(S) \rightarrow X$$

$$\neg ex(S) \rightarrow Y$$

$$\neg ex(S) \rightarrow Z.$$

Now, take for modeling purposes:

$$\neg ex(S) \rightarrow Y \text{ namely, "Sensors often dislike complications".}$$

How do we interpret "complications"? In our case, it refers to "complicated methods". So, we understand the above sentence as follows:

Oftenly, If S is a sensing learner, and X is a method, and X is complicated, then it is not the case that S likes to learn X .

Hence, the last sentence can be represented as:

```
- like_toK(S, X) :- sensing_learner(S),
                    complicated(X),
                    method(X),
                    not ex(S).
```

In our context of ASP, "–" is the strong negation operator, and "not" is the default negation operator. Sometimes we also use \neg for "not".

In conversations cognitive elements are reflected through logical formulas. For example, by detecting a student with a profile prone to anxiety and a superficial learning approach, the DLV software traces routes to offer the student a variety of exercises based on mindfulness and other techniques that are related to their learning style, with the intention of helping the user to facilitate their state of receptivity and favor the learning process. The student receives constant feedback to redefine their characteristics and preferences to be used in their next interaction.

The main contribution of our paper is the following: As far as the authors know, this is the first paper to explore the use of *Answer Set Programming* (and in particular DLV) to define an intelligent agent (in collaboration with Python) to generate automatic class notes considering the learning styles of the students. Furthermore, unlike most herding literature, the paper employs (enhances) dialogs to motivate the students.

These enhanced dialogs include a versatile conversation (with emphasis in the use of mindfulness).

The paper is structured as follows: In the next section, we start by stating the general background of the paper. Of particular interest we point out the use of *Answer Set Programming* to model our logical agent. We provide in Section 3 an example of the kind of dialogs that our system could generate. In Section 4 we present the main contribution of our paper, namely the analysis, design and implementation of our system. We conclude the paper with a brief discussion of related work and the achieved results.

2 Background

In this section, we first present the general background of Artificial Intelligence from this article.

Then we continue to present the psychological/pedagogical background that we consider relevant for the design of our system.

2.1 Background

Computer Science and Artificial Intelligence have demonstrated and reached a proper level of development in which it is possible to achieve our main objective.

The scientific goal of *Artificial Intelligence (AI)* [4] is to comprehend intelligence by building software that reveals intelligent comportment. AI deals with

the perceptions and approaches of figurative inference, or reasoning, by a software system, and how can the knowledge be used to create those inferences will be embodied in the machine. The expression intelligence deals with many cognitive skills such as understand natural languages, the ability to solve problems, and learn.

Recently, AI although is usually misunderstood by popularity as machine learning using deep learning, AI is by far more than this. AI has growth in several branches as science such as *Knowledge Representation (KR)*, *Expert Systems (ES)*, *Natural Language Processing (NLP)*, *Logical Reasoning (LR)*, among others.

Around 1960, John McCarthy first proposed the use of logical formulas as a basis for a knowledge representation language of this type. However, it was necessary to develop new semantics under logic programming languages to develop *Common Sense Reasoning* such as *Answer set Programming (ASP)*. *Knowledge representation (KR)* is one of the most important subareas of artificial intelligence [5]. If we want to design an entity (a machine or a program) capable of behaving intelligently in some environment, then we need to supply this entity with sufficient knowledge about this environment. To do that, we need an unambiguous language capable of expressing this knowledge, together with some precise and well-understood way of manipulating sets of sentences of the language which will allow us to draw inferences, answer queries, and to update both the knowledge base and the desired program behavior.

ASP [1] is a logic programming formalism for knowledge representation and reasoning that posses the following suitable properties:

1. Based on different from classical logic, allows nonmonotonic reasoning: the absence of beliefs can be used to make inferences, and the addition of beliefs can prevent inferences. This allows the usage of ASP for defining default inferences that can be blocked once more information becomes available [6].
2. It has three fundamental characteristics [7]: a modeling language based on the syntax of logic programs, the use of the answer set semantics to interpret programs in that language, and a problem-solving methodology in which a program is written, so that its answer sets provide solutions.
3. It is a well known AI paradigm for decision-making and problem solving. It has proven useful in a variety of application areas [8] [1] [9], such as Biology, Psychology, Medicine and music composition. ASP is a declarative programming language used to specify a problem in terms of general inference rules and constraints, along with concrete information about the application scenario.
4. It is a prominent knowledge representation and reasoning paradigm that found both industrial and scientific applications. The success of ASP is due to the combination of two factors: a rich modeling language and the availability of efficient ASP implementations [9].
5. It is in close relationships to other formalisms such as *Propositional Satisfiability*, *Satisfiability Modulo Theories*, *Quantified Boolean Formulas*, *Constraint Programming*, *Planning*, *Scheduling* and many others [9].

A robust and efficiency-oriented first implementation of an ASP system called DLV (Disjunctive Logic Programming) was made only available in 1997 after 15 years of theoretical research on ASP. The development of DLV was started in 1996 in a research project funded by the Austrian Science Fund. Currently DLV is the subject of an international collaboration between the Vienna Technical University and the University of Calabria [10]. After its first release, the DLV system has been significantly improved over and over in the last years, and its language has been enriched in several ways. Relevant optimization techniques have been included in all DLV engine modules, containing database techniques for efficient instantiation and novel techniques for response Set Verification, Heuristics and Advanced Section Operators for the Model Generation. As a result, currently DLV is generally accepted as the State of the Art. It is widely used by researchers around the world and it is also competitive from the point of view of efficiency with the most advanced systems in the field of ASP[10].

In this paper we re-used and adapted a general architecture which is detailed in [11], [12], [13] and [14]. It is worth to mention that such architecture was originally designed for the development of an intelligent chat-bot agent. But since such architecture was aimed to chat with a student and since our textnotes are presented as a hypothetical dialog between a student and a *Teacher Assistant (TA)*, we found feasible to re-use several ideas of such system (considering of course some adjustments).

2.2 Psychopedagogical Considerations

The constructivist theory of education postulates that the subject builds his knowledge based on contexts that are meaningful to him. In such a way that the human being is conceived as an active agent of knowledge, making interpretations of the world that surrounds him in order to adapt to social reality [15]. Learning must include the exercise of what is learned, conceptual knowledge and the learning context, which is inserted in the culture. From this perspective, the subject is an active element, in charge of manufacturing and deciphering the information it receives. [16]. With each new interaction of the subject with his environment, mental structures are developed that allow him to unite the past knowledge with the new [17]. In the construction and orientation of these structures, the presence of a guide becomes essential, who can be the educator, virtual or physical. This is in charge of planning the learning processes depending on two levels of development [18]:

- 1) Real, that refers to what the subject is capable of doing without help.
- 2) Potential, that indicates what the person is capable of achieving with the help of mediating elements.

For the identification of these elements it is important that the educational staff make use of tools that allow them to place the subject not only in the levels of development, but also contribute so that the subject manages to control the ways in which they learn. We have the classification of the learning styles have been provided by Felder and Spurlin who describe each category in

one of the four dimensions: sensing/intuitive, visual/verbal, active/reactive and sequential/global [19].

Although in some currents in psychology the scientific evidence that learning styles significantly improve learning itself is questioned (Tesia Marshik, to mention one of these), this work assumes, without granting, that student emotional situation influences, directly or indirectly, in their learning.

The well accepted constructivist theory aims to develop socialized learning that allows the subject to be formed not only at the intellectual level but also at the psycho-affective level, using knowledge to build the reality in which it is situated. The use of technological tools must be adapted to the student, making modifications to adapt to the user's needs. Baker mentions that in the specific case of the use of technological support systems in education, constant feedback is required to allow the system to be constantly improved [20].

3 Motivation

We present a short and incomplete example of notes generated for Laura, a freshman college student whose major is literature. She is a student with a considerable level of anxiety. Laura(of sensitive learning style) declares a lack of understanding in her class. The name of the student character is also Laura, a decision taken on purpose by the system. Before the interaction, each student already provided personal and academic information, with the intention of offering a personalized intervention.

Draco: Hi. Before starting, I would like to introduce myself. I am something similar to a Teacher Assistant, you can call me Draco, my task will be to accompany you in your learning processes, offering you some techniques that could help you a better understanding of some ideas of the course, and perhaps increase your learning skills. But I will also be here to support you when you feel tired, stressed, anxious, sad, or simply when you need a calmer state of mind. I would like to ask you to be patient with me, since I am also learning and your help will be very useful to me to improve. What is your name?

Laura: My name is Laura

Draco: Nice name Laura. Your name reminds me Laura Maria Caterina Bassi. She was an Italian physicist and academic born in 1711. Recognised and depicted as "Minerva" (goddess of wisdom), she was the second woman in the world to earn the degree of Doctor of Philosophy (after the philosopher Elena Cornaro Piscopia, who had received doctorate in 1678) and the first woman to have doctorate in science. You know what? she was also the first salaried woman teacher in a university. In fact, at one time she was the highest paid employee. Even more, she eventually became the first university professor in the world. Imagine that she was also the first woman member of any scientific establishment, when she was elected to the Academy of Sciences of the Institute of Bologna in 1732. How great!

Laura: How and where was Laura educated?

Draco: Bassi had no formal education and was privately tutored from age five till she was twenty. By then she was well versed in all major disciplines including sciences and mathematics. Noticing her ability, Prospero Lambertini, the Archbishop of Bologna (later Pope Benedict XIV) became her patron. With Lambertini's arrangement she publicly defended forty-nine theses before professors of the University of Bologna on 17 April 1732, for which she was awarded a doctoral degree on 12 May. A month later, she was appointed by the university as its first woman teacher, albeit with restriction to teach all-male classes. Lambertini, by then the Pope, helped her to receive permission for private classes and experiments, which were granted by the university in 1740. Bassi became the most important populariser of Newtonian mechanics in Italy.

Draco: I learn this information from Wikipedia. I love it!

Draco: But tell me Laura, how do you feel today? Is there something you would like to tell me?

Laura: I do not feel well, I feel anxious and frustrated because I cannot understand the class, I am very slow to learn and I despair.

Draco: I understand, that must feel very overwhelming.

We interrupt our example here.

4 Class Notes Intelligent Agent System Design

The objective of the system is to generate logic text notes (in the form of a dialogue) for the student. It is very important to bear in mind that the dialogue constructed by the agent is staged by a hypothetical student that we hope reflects the same real student.

4.1 Brief Analysis

AI approaches to problem-solving and decision-making are becoming more and more complex, leading to a decrease in the understandability of solutions [21]. The European General Data Protection Regulation (GDPR) have tried to tackle this problem by stipulating a "right to explanation" for decisions made by AI systems. It states that anyone has the right to reject a "decision based solely on automated processing" that "significantly affects" a person. This new regulation may not come as a surprise since most Artificial Intelligence methods are "black-boxes", that is, they produce accurate decisions, but without the means for humans to understand why a decision was computed [21]. This is specially observed since the popularity of the use of machine learning as AI production of software. Machine learning approaches uses neural networks (among others) where the designed software are learning based on training examples but there is no way to understand the knowledge learned in an explicit way. Since the neural network is a mathematical structure framework based on probabilities, machine learning works as learning of patterns by behaviorism to produce other patterns.

However ASP as a Knowledge Representation/Reasoning programming language has the advantage of expressing semantical expert knowledge in an explicit

way in such a way that it is possible to control the knowledge whenever an AI agent expert system is being designed, as well as the inferences of this knowledge and the knowledge that causes such inferences [21]. Here there is a wide spread distance from machine learning for instance. A Machine Learning System (MLS) could learn by training of examples the knowledge of expressions in a community but a MLS is not able to control the type and semantics of the learned knowledge. Cases have been reported where the Machine Learning Agents have learned imprecise behaviors when they are trained with a broad amount of examples, trying to model great amount of knowledge. On the other hand, ASP allows structurally- to control the semantic aspects that significantly improves the translation of problem stated in human terms, to offer AI solutions within a context that may even contain ethical aspects within the model. These considerations make the use of ASP plausible in the modelling of a family of problems, including those discussed in this article. We highlight four predominant elements

1. An agent must have extensive knowledge of the domain in which it must act, and on its own capacities and objectives. The domain is pedagogy and psychology as well as the profile of the student.
2. It should be able to frequently expand this knowledge by new information coming from observations, communication with other agents, and awareness of its own actions. Communication in our case is carried out with the student periodically asking him to fill out surveys.
3. All this knowledge cannot be explicitly represented in the agent’s memory. This implies that the agent should be able to reason, i.e. to extract knowledge stored there implicitly.
4. Finally, the agent should be able to use its knowledge and its ability to reason to rationally plan and execute its actions.

Since the above observations assumes a solid theoretical foundations of agent design, a robust proposal should be based on theories of knowledge representation and reasoning. Two well known theories that we consider are: The theory of logic programming and nonmonotonic reasoning [22], [23], [24], and the theory of actions and change [25].

4.2 Design

The general design of our system is presented in this section. We follow a general design of an intelligent agent as proposed in [1]. Furthermore we consider in addition the work presented in [11–13].

Our agent follows the standard observe-think-act loop: It first senses the environment. In our case it asks the student to respond a survey. Then it loops as follows:

1. An action selects a target, which is recorded in the agent’s memory. Our target is nothing more than the topic of a concrete enhanced lecture namely, a lecture in the form of a dialogue between a TA and a student with a

psychological component. The agent uses a theory of intentions along with background information in order to come up with a plan to achieve this goal and form the intention to do it. The agent executes components of the expected activity and keep the history of these executions along with the observation what could have been done during this process.

2. It sense again the environment. In our case it asks again the student to respond a new short survey.

Student model The system assumes a student model that is almost empty at the beginning but is updated thanks to the interaction between the student and the system. This model is constructed based on the preferences and needs of the student. Having a user model is fundamental in this kind of personalized systems as we have learned and adjusted to our particular case from the presented in [26].

A Master-Slave AI design We propose to follow a master-slave conceptual design following a centralized approach as defined in [11]. Namely, we create hundreds of slaves (at least one thousand) such that each of them can perform a very concrete task. All the tasks correspond to interactions with our e-student. An example could be as simple as ask the e-student the definition of "an interpretation", or just perhaps to congratulate the student for a particular reason. A more complex task could be to progressively teach the student simple mindfulness exercises or meditations exercises. Associated to each slave we have its semantic knowledge. For instance, slave named *E3* could correspond to an exercise of sound-mindfulness, that belongs to the set of mindfulness exercises. Further more, the system has an explicit logical rule saying that this type of exercise normally helps to reduces anxiety, and so on. Note that these default rules can naturally be expressed in ASP and are very useful in this context. All the semantic knowledge of each slave plus a general theory of interaction among them is written in DLV. The system can also be seen as an approach to automate use of basic software component libraries. This reminds to the factory strategy with the aim of making automated software composition [27].

The Intelligent Agent System At the beginning of the semester a student is required to fill out a questionnaire. The student also provides feedback on the clarity of the questionnaire. The survey consists of the predominant learning style(s) of the student. In addition, it includes questions about the general profile of the student as well as other questions regarding the emotional status of the student. The system is a *Reasoning Planning System* consists of a cycle of 4 sequential processes-modules described below.

I. **Abstract Script Dialogue Session (ASDS)**: The ASDS basically consists of two modules of KB-reasoning represented and specified via ASP, the lowest one consists of a logical theory (explained in further detail through the next section) that generates a set of recommendations (resources/assets) that would correspond to an abstract plan. The highest module consist of an ASP

program that proposes the ASDS plan solving an specific problem based in its logical theory of actions.

The problem is to compose a dialogue session as a sequence of tasks such that it tries to improve the knowledge of the e-student about the given subject. At the same time, it is intended that the session offers alternatives to the e-student to relax the emotional state and help him to go through a series of exercises, reflections and activities that offer added value to the session. Do not forget that we refer here to the character of the class notes, not the real student. However, we expect the student would become identified with his/her character.

We wrote the formal definition of a dialogue session using the standard generate and test approach in ASP. Our definition is the following:

```

assignFinal(I,Y) :-      nOrden(I), rrr(Y),
                        not thereIsAnotherOne(I,Y).
thereIsAnotherOne(I,Y) :- nOrden(I), rrr(Y), rrr(Y1), Y!=Y1,
                        assignFinal(I,Y1).
:- assignFinal(I,S), assignFinal(J,S), I != J.

```

where rrr is defined as:

```

rrr(R) :- resource(R), open(R), not yetUsed(R).

```

The following is an just an example a logical rule in our theory of learning styles .

```

%Sensor students often like solving problems by well-established methods
like_toK(S,s(X,M)) :- sensing_learner(S),
                       method(M),
                       type_of_method(M, well_est),
                       problem(X),
                       can_be_applied(M,X),
                       not ex(S).

```

II *Generating BSRL code*: The idea behind the code that we call for reference as BSRL code is to define a basic programming language such that any program of our library is a highly malleable object where one could define some operators (such as mutation, crossover composition, selection, specialization, generalization). Two basic examples are the following Briefly speaking, each instruction in the BSRL code is a triple $\langle l, o, a \rangle$ where l is a label, o an operator and a is an argument. BSRL resembles a kind of machine assembly language. Details can be obtained from [14].

III. *Generation of the class notes (as a dialogue)*: The Dialogue Module corresponds to the director of the orchestra that executes the composed dialogue session as interactions of AI-Task with the e-student.

IV. *Feedback Extraction of Relevant Information and Knowledge Module*: We have two kinds of feedbacks: Real and Virtual

Real One: We apply a questionnaire, open and available to anyone who wants to answer it, that provides feedback on the clarity of the class notes, as well as eight other set of questions on his/her new mental state.

Virtual One: This is the feedback that we are concern in our AI agent. This module filters and updates the questions/answers of the dialogue between the TA and the student characters. Recall that our system tries to simulate a possible real conversation between the real student and the TA. For instance, if the survey of the student shows interest in a given topic, the conversation address such topic and the student character also shows interest in this topic. If necessary it might uses random outcomes.

4.3 Implementation (Pipeline)

It is standard to write EDB to denote an extensional database, namely the facts of a logic program. EDB_i denotes the initial EDB. EDB_x denotes the current EDB. Let T be the theory (or logic program knowledge base) as described in the previous subsection.

A): The student is asked at the beginning of the course to fill out the initial survey (that we call S_i).

B) We execute a basic python program that transforms S_i into a file of DLV facts, that we call EDB_i. Let EDB_x be EDB_i.

We loop steps (C, D, E) until the semester is over:

C): We use EDB_x as the input to execute our software to generate our notes (for two weeks). So, given the theory T, we compute the answer sets of T union EDB_x.

D): After the two weeks of classes we ask the student to fill out the follow-up survey (that we call S_f).

E) A simple python program is executed to convert S_f into a file of DLV facts, that we call EDB_u. Let EDB_x be EDB_x union EDB_u.

We respect to C) we use a unix shell file to execute DLV + Python to obtain our intended class notes.

5 Results, conclusions and Future work

Among the various applications of this work, we will mention that in addition to the proposed application of creating notes that are appreciated by the reader for adapting to their learning style, it can also serve as a "bridge" between the styles that prevail within a radius of influence of the reader. It is ambitious to assume that each person can be characterized within a learning style. It is reasonable to assume that each person is an amalgam of learning styles that can even vary according to volatile moods. These personalized notes may offer a range of possibilities, according to the knowledge bank and the interaction with the reader.

Some potential applications of the project we will mention the following:

-The use of the notes is not restricted to the use of the reader, also for the academic faculty, not only for the teachers of the subject itself, but for subjects in which the notes are interconnected: history, literature and philosophy, among others. This has the advantage of presenting the knowledge and intellectual development of humanity as a single entity connected in conducting threads that are bidirectional. For teachers of a subject, it can help them to understand the philosophy of their colleagues - after all, an instructor's marks largely reflect their own teaching philosophy. It is possible to think from a pedagogical point of view, that what is important about the notes is not in what they say, but in what they stop saying. This is a key point in creating instructor notes, due to obvious space limitations, the choice of perspective can be appreciated and enriched by your academic peers.

-Student knowledge is never overrated. This is possibly the reason for evaluating them at regular intervals of time, by means of midterm exams, midterms, etc. This in order to understand their learning process and make global adjustments to the course for better understanding. However, global adjustments frequently ignore the emotional state of the student as a critical element in their learning process. The knowledge bases will allow to know more between the students and the generations of students and their psycho-academic evolution.

For future work we consider to explore the idea of representing Knowledge using alternative non-monotonic paradigms (besides from ASP) such as those found in [28–33].

References

1. Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press (2014), ISBN 9781107782877
2. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. *ACM Trans. Comput. Log.* **2**(4), 526–541 (2001), <https://doi.org/10.1145/383779.383783>
3. Osorio, M., Pérez, J. A. N., Arrazola, J.: Equivalence in Answer Set Programming. In: A. Pettorossi, editor, Logic Based Program Synthesis and Transformation, 11th International Workshop, LOPSTR 2001, Paphos, Cyprus, November 28-30, 2001, Selected Papers, vol. 2372 of *Lecture Notes in Computer Science*, pp. 57–75, Springer (2001), https://doi.org/10.1007/3-540-45607-4_4
4. Abu-Nasser, B.: Medical expert systems survey. *International Journal of Engineering and Information Systems (IJEAIS)* **1**(7), 218–224 (2017)
5. Baral, C., Gelfond, M.: Logic programming and knowledge representation. *The Journal of Logic Programming* **19**, 73–148 (1994)
6. Schüller, P.: Answer Set Programming in Linguistics. *KI-Künstliche Intelligenz* **32**(2-3), 151–155 (2018)
7. Denecker, M., Vennekens, J., Bond, S., Gebser, M., Truszczyński, M.: The second answer set programming competition. In: International Conference on Logic Programming and Nonmonotonic Reasoning, pp. 637–654, Springer (2009)
8. Lifschitz, V.: Answer set programming. Springer International Publishing (2019)
9. Gebser, M., Leone, N., Maratea, M., Perri, S., Ricca, F., Schaub, T.: Evaluation Techniques and Systems for Answer Set Programming: a Survey. In: *IJCAI*, vol. 18, pp. 5450–5456 (2018)

10. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. *ACM Transactions on Computational Logic (TOCL)* **7**, 499–562 (2002), <https://doi.org/10.1145/1149114.1149117>
11. Osorio, M., Zepeda, C., Castillo, H., Cervantes, P., Carballido, J. L.: My University e-Partner. In: *CONTIE*, pp. 150–1503 (2019)
12. Osorio, M., Zepeda, C., Carballido, J. L.: Towards a virtual companion system to give support during confinement. *CONTIE* (2020)
13. Osorio, M., Zepeda, C., Carballido, J. L.: MyUBot: Towards an artificial intelligence agent system chat-bot for well-being and mental health. Accepted in *Artificial Intelligence for HEaLth, PersonaLized MedIcine aNd Wellbeing Workshop at ECAI 2020*. (2020)
14. Osorio G., M., Montiel M., L. A.: E-friend: A logical-based AI Agent System Chat-bot for Emotional Well-being and Mental Health. Accepted in *DeceptECAI 1st International Workshop on Deceptive AI at ECAI 2020*. (2020)
15. Guerra Garcia, J.: El aporte de la teoria sociocultural de Vygotsky para comprender la construccion del conocimiento en el ser humano. *Dilemas Contemporaneos: Educacion, Politica y Valores* **7**(2) (2020)
16. Martin, G. M., Martinez, R. M., Martin, M. M., Nieto, M. I. F., Nunez, S. V. G.: Acercamiento a las Teorias del Aprendizaje en la Educacion Superior. *Revista UNIANDES Episteme* **4**(1), 48–60 (2017)
17. Saez, M. R.: La educacion constructivista en la era digital. *Revista Tecnologia, Ciencia y Educacion* (12) (2019)
18. Cruz, C., Lorenzo, F., Hernandez, P.: La obra de Vygotsky como sustento teorico del proceso de formacion del profesional de la educacion primaria. *Conrado* **15**, 600–614 (2019)
19. Felder, R. M., Spurlin, J. E.: Applications, Reliability, and Validity of the Index of Learning Styles. *Intl. Journal of Engineering Education* **21**, 103112 (2005)
20. Baker, R.: Stupid Tutoring Systems, Intelligent Humans. *International Journal of Artificial Intelligence in Education* **26**, 600–614 (2016)
21. Fandinno, J., Schulz, C.: Answering the why in answer set programming—A survey of explanation approaches. *Theory and Practice of Logic Programming* **19**(2), 114–203 (2019)
22. Baral, C., Gelfond, M.: Logic Programming and Knowledge Representation. *J. Log. Program.* **19/20**, 73–148 (1994), [https://doi.org/10.1016/0743-1066\(94\)90025-6](https://doi.org/10.1016/0743-1066(94)90025-6)
23. Lifschitz, V.: Foundations of declarative logic programming. In: G. Brewka, editor, *Principles of Knowledge Representation*, CSLI Publications (1996)
24. Marek, V. W., Remmel, J. B.: On the Foundations of Answer Set Programming. In: A. Proveti, T. C. Son, editors, *Answer Set Programming, Towards Efficient and Scalable Knowledge Representation and Reasoning*, Proceedings of the 1st Intl. ASP’01 Workshop, Stanford, CA, USA, March 26-28, 2001 (2001)
25. Gelfond, M., Lifschitz, V.: Action Languages. *Electron. Trans. Artif. Intell.* **2**, 193–210 (1998)
26. Baskar, J., et al.: Cognitive architecture of an agent for human-agent dialogues. In: *Practical Applications of Agents and Multi-Agent Systems*, pp. 89–100, Springer (2014)
27. Greenfield, J., Short, K.: Software factories: assembling applications with patterns, models, frameworks and tools. In: *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pp. 16–27, ACM (2003)

28. Osorio, M., Nieves, J. C.: PStable Semantics for Possibilistic Logic Programs. In: A. F. Gelbukh, A. F. K. Morales, editors, MICAI 2007: Advances in Artificial Intelligence, 6th Mexican International Conference on Artificial Intelligence, Aguascalientes, Mexico, November 4-10, 2007, Proceedings, vol. 4827 of *Lecture Notes in Computer Science*, pp. 294–304, Springer (2007), https://doi.org/10.1007/978-3-540-76631-5_28
29. Osorio, M., Jayaraman, B.: Aggregation and Negation-As-Failure. *New Gener. Comput.* **17**(3), 255–284 (1999), <https://doi.org/10.1007/BF03037222>
30. Osorio, M., Jayaraman, B., Plaisted, D. A.: Theory of Partial-Order Programming. *Science of Computer Programming* **34**(3), 207–238 (1999)
31. Nieves, J. C., Osorio, M., Zepeda, C.: A Schema for Generating Relevant Logic Programming Semantics and its Applications in Argumentation Theory. *Fundam. Informaticae* **106**(2-4), 295–319 (2011), <https://doi.org/10.3233/FI-2011-388>
32. Nieves, J. C., Cortés, U., Osorio, M., Olmos, I., Gonzalez, J. A.: Defining new argumentation-based semantics by minimal models. In: Seventh Mexican International Conference on Computer Science, ENC 2006, 18-22 September 2006, San Luis Potosí, Mexico, pp. 210–220, IEEE Computer Society (2006), <https://doi.org/10.1109/ENC.2006.10>
33. Confalonieri, R., Nieves, J. C., Osorio, M., Vázquez-Salceda, J.: Possibilistic Semantics for Logic Programs with Ordered Disjunction. In: S. Link, H. Prade, editors, Foundations of Information and Knowledge Systems, 6th International Symposium, FoIKS 2010, Sofia, Bulgaria, February 15-19, 2010. Proceedings, vol. 5956 of *Lecture Notes in Computer Science*, pp. 133–152, Springer (2010), https://doi.org/10.1007/978-3-642-11829-6_11