HEALTHY NUTRITION UNDER ASP-PROLOG

Fernando Zacarias F.[1] Rosalba Cuapa[2], Erick Madrid[1], Dionicio Zacarias[1]

[1]Computer Science, BUAP, Puebla, Pue., México
fzflores@yahoo.com.mx

[2]Faculty of Architecture, BUAP, Puebla, Pue., México
rcuapa_canto@yahoo.com

ABSTRACT

A child's diet and his nutrition is a growing concern in Mexico. Obesity rates triple in last 3 decades in our country. Thus, a novel and attractive tool designed for youth is deployed. It has as main objective reduce this problem in a way friendly and intelligent. It includes a planning system based on propositional logic named DLV^K and a novel architecture that allows us to extend our solution to mobile devices. In this proposal we consider goal-based agents that use more advanced structured representations and are usually called planning agents. These agents are supposed to maximize their performance measure. Achieving this is sometimes simplified if the agent can adopt a goal and aim at satisfying it. Thus, first we present the well-known problem called The Wumpus world. Second, present an application to assist in the solution of Nutrition in Mexico. Planning for nourishing using DLV^K implemented for mobile devices through the WAP protocol using GPRS service is deployed. Both problems viewed as a logic program (into DLV^K) whose answer sets corresponds to solutions. Plans correspond to answer sets for these programs, in the spirit of answer set programming.

KEYWORDS

Planning, Logic programming, Mobile applications, Health & AnsProlog

1. INTRODUCTION

Planning is a key ability for intelligent systems, incrementing their autonomy and flexibility. When we think in intelligent systems is common to think of making use of intelligent agents that allow us to model problems in planning. Planning means the construction of sequences of actions to achieve their goals. In the last three decades it has been an area of research in artificial intelligence. Planning techniques are applied in a variety of task including robotics, process planning, web based information gathering, autonomous agents, spacecraft mission control and actually in workflow construction on the computational Grid. Planning involves the representation of actions and world models, reasoning about the effects of actions, and techniques for efficiently searching the space of possible plans. There are several logic-based languages for reasoning about actions and action planning, such as: [1], [2], [3] and [4]. In this paper we use the DLV^K system to create formal model of the effects of actions on the world. It is based on a new language called K (where K should remind of state of knowledge) for planning [5] and [6]. This language is very flexible, and is capable of modeling transitions between states of the world (i.e., states of complete knowledge) and reasoning about them as a particular case. K is closer in spirit to answer set semantics [7] than to classical logics. It supports the explicit use of default negation, and thus exploiting the power of answer sets to deal with incomplete knowledge. K is implemented as a front-end on top of the DLV answer set programming system [8].

The paper is organized as follows. In section II briefly introduce one of the well-known planning problems in artificial intelligence showing how this can be modeled in DLV^K. Our proposed

DOI : 10.5121/ijcnc.2013.5307

solution to the real nutrition problem, that is facing our country, is presented in section III. Section IV is concerned to mobile architecture used to carry our handy solution for anyone. Finally, in section V we give some concluding remarks and discuss further plans.

2. PLANNING PROBLEMS IN DLV^K

In this section, we describe how planning problems can be described as “programs” in DLV^K system. In this context, we use an example in the well-known the Wumpus world domain. DLV^K programs are built using statements of the language K, plus further optional control statements. The representation of this problem can be in a similar form as the formal definition used in [9].
A planning problem is a pair $P = <PD, q>$ of a planning domain *PD* in a query *q*, which specifies the goal. A planning problem is represented as a combination of a background knowledge, which is represented by a rational agent, and a program of the following general form:

fluents:	F_D
actions:	A_D
Initially:	I_R
always:	C_R
goal:	q

Where fluent represent basic properties of the world, which can change over time. They are similar to propositional assertions. States are collections of fluents, each of them is associated with a true-value. We distinguish between so called *world states* and *knowledge states*. The current state of the world with respect to a set of fluent is: $F = \{f_1, \ldots, f_n\}$, can be defined as a function $s : F \rightarrow$ {true, false}, that is, a set of literals which contains either f nor $\neg f$ for any $f \in F$. s is a state of knowledge. From an agent’s point of view, states can also be seen as partial functions s’, that is, consistent sets of fluent literals, where for a particular fluent $f \in F$ neither f nor $\neg f$ may hold.

2.1. The Wumpus world in DLV^K

The Wumpus world is a cave consisting of rooms connected by passageways. Lurking somewhere in the cave is the wumpus, a beast that eats anyone who enters its room. The wumpus can be shot by an agent, but the agent has only one arrow. Some rooms contain bottomless pits that will trap anyone who wanders into these rooms (except for the wumpus, which is too big to fall in). The only mitigating feature of living in this environment is the possibility of finding a heap of gold. Although the wumpus world is rather tame by modern computer game standards, it makes an excellent testbed environment for intelligent agents. A sample wumpus world is shown in Figure 1a) and 1b). In this example, we define the fluents that describe the Wumpus world as follows:

fluents:

```
    start(X2,Y2) requires world(X2,Y2).
    hunter(X3,Y3) requires world(X3,Y3).
    wumpus(X,Y) requires world (X,Y).
    position_stench(X1,Y1) requires world (X1,Y1).
    position_pit(X4,Y4) requires world (X4,Y4).
    position_breeze(X5,Y5) requires world(X5,Y5).
    position_gold(X6,Y6) requires world (X6,Y6).
    golddown.
    Hunterlive
```

The actions defined for Wumpus world correspond to:

actions:

```
        holdgold.
        moveright costs 1.
        moveleft costs 1.
        moveup costs 1.
        movedown costs 1.
        dead.
```

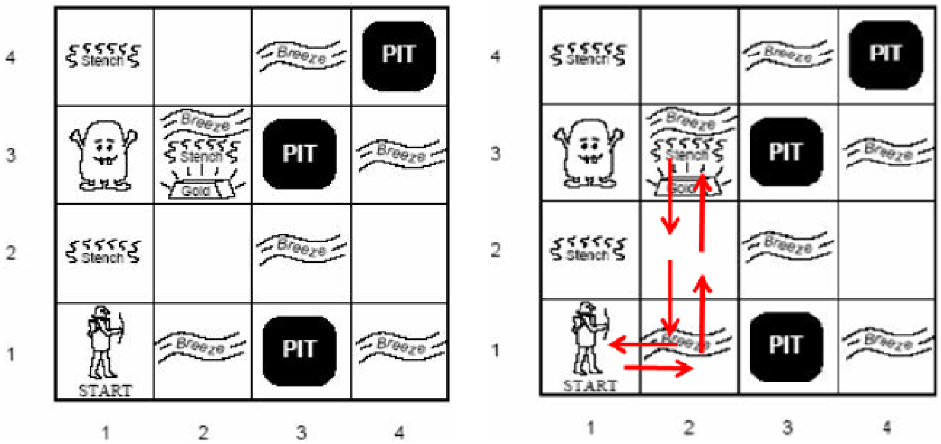


Figure 1a) A typical Wumpus world 1b) Movements necessary for the plan

Actions may have assigned costs, which should be minimized. In our case, each move has costs 1, resulting in plans, where a minimum number of moves is executed to achieve the plan. The rules following the declarations of actions and fluents describe the transitions and constraints on the initial states of the domain (for space reasons we do not present all code). Finally, the goal: section defines the goal to be reached and the plan length.

```
always:     executable holdgold if position_gold(X,Y),hunter(X,Y).
            caused -golddown after holdgold.
            caused hunter (X,Y) after holdgold, hunter (X,Y).
            caused hunterlive after holdgold, hunterlive.
            caused -hunterlive after holdgold, - hunterlive.
            executable moveright if hunter(X,_), X<4.
            caused hunter(R,Y) after moverright, hunter(X,Y), add(X,1,R).
            caused golddown after moveright, golddown.
            caused -golddown after moveright, -golddown.text,
```

Next we specify when an action moveright is executable. The first condition states that this is possible if hunter is at position X and it is less than 4, i.e., if the hunter is before the wall. The following rules with the keyword "caused" define the effect occurred after moveright. For example, in first rule, hunter advanced to (R, X) position after having added one position to the starting position, verifying forward movement is possible and not bumps into the wall.

```
        inertial wumpus(X,Y).            inertial position_stench(X1,Y1).
        inertial position_pit(X4,Y4).    inertial position_breeze(X5,Y5).
        inertial position_gold(X6,Y6).   inertial start(X2,Y2).
        noConcurrency.                   initially:
        % initial configuration to the wumpus world
```

```
start(1,1).          hunter(1,1).      position_gold(2,3).       wumpus(1,3).
position_stench(1,2).                  position_stench (1,4).             position_stench    (2,3).
posicion_pit(3,1).                     posicion_pit(3,3).                 posicion_pit(4,4).
posición_breeze(2,1).                  posicion_breeze(4,1).              posicion_breeze(3,2).
posicion_breeze(2,3).                  posicion_breeze(4,3).              posicion_breeze    (3,4).
golddown.                              hunterlive.
```

goal: -golddown, hunter(1,1), hunterlive ? (7).
The rule goal, defines the goal to be reached and the plan length.

2.2. Running the Wumpus world in DLV^K

Assume that the above background knowledge and planning program are giving in files wumpus.bk and wumpus.plan, respectively. The execution of the command:

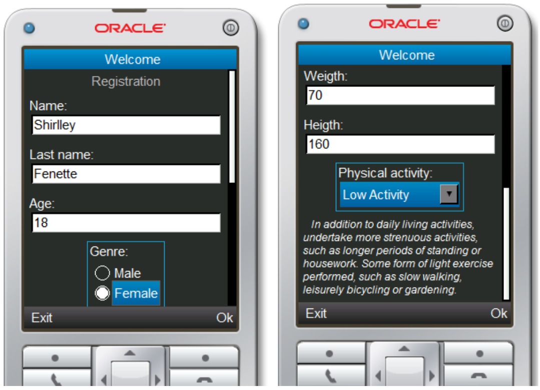
```
C:\ dlv wumpus.bk wumpus.plan –FP –n=1
```

Computes the following result: **PLAN:** moveright; moveup; moveup; holdgold; movedown; movedown; moveleft; COST 6

As you can see, this plan is the minimum found (as you can see in figure 1b. Where plan steps are separated by ";" The command-line option –n=1 tell dlv to compute only one plan. Option –FP is used to invoke the planning front-end in dlv. Alternatively, one can use the options –FPopt and FPsec, which explicitly enforce optimistic and secure planning, respectively. For planning problems with a unique initial state a deterministic domain, these two options do not make a difference, as optimistic and secure plans coincide in this case. As you can see, this solution is clear, simple and all rules in IR and CR have to satisfy the safety requirement for default negated type literals, i.e., each variable occurring in a default negated type literal has to occur in at least one non negated type literal or dynamic literal.

3. SOLVING REAL PROBLEMS OF PLANNING DIETS

Almost all countries are experiencing an obesity epidemic. In developed countries, obesity is not only common in the middle age, but is becoming ever more common among younger adults and children. The fact is that a poor diet may have an injurious impact on health. There is a need to provide a safe, evidence-based and easy-to-access dietary advice for everyone, especially children. Thus, we present a novel mobile application that allows us to obtain a custom and automatic plan to support people improve their general health.



Name: Shirlley

Last name: Fenette

Age: 18

Genre: ○ Male ● Female

Weigth: 70

Heigth: 160

Physical activity: Low Activity

In addition to daily living activities, undertake more strenuous activities, such as longer periods of standing or housework. Some form of light exercise performed, such as slow walking, leisurely bicycling or gardening.

Figure 2 Representing personal data

Next, we show (in figure 2) the interface of the mobile application developed to gather all the necessary information before computing any plan (age, weight, height and physical activity). Those are needed to calculate the BMI (Body mass index is defined as the individual's body mass divided by the square of their height) and check if this is between the normal range according to the Table 1; and estimate the basal energy expenditure (B.E.E) using the Harris-Benedict equations [11]. The B.E.E is multiplied with the physical activity value (1 to 2) to calculate the calorie requirement per day (kcal/day) as in the example shown in figure 3a).

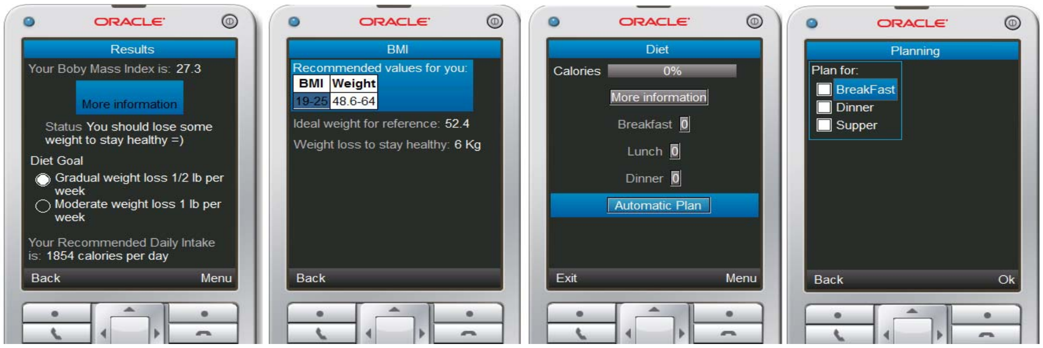
Now, we can compute the plan to reach the calorie requirement (55% carbohydrates, 15% proteins, and 30% fats) divided into 3 meals: breakfast, lunch and dinner. These are separated to reach the goal to re-plan one or two if the plan is manually changed by the user. (See figure 3b).

Table 1. Project selection matrix rules

Age	**Man**	**Woman**
5	13 – 16.7	12.7 – 17
6	13 – 17	12.7 – 17.3
7	13.2 – 17.4	12.7 – 17.7
8	13.3 – 17.9	12.9 – 18.3
9	13.5 – 18.4	13.1 – 19
10	13.7 – 19.1	13.5 – 19.8
11	14.1 – 19.9	13.9 – 20.7
12	14.5 – 20.8	14.4 – 21.7
13	14.9 – 21.7	14.9 – 22.7
14	15.5 – 22.6	15.4 – 23.5
15	16 – 23.5	15.9 – 24.1
16	16.5 – 24.2	16.2 – 24.5
17	16.9 – 24.9	16.4 – 24.8
18	17.3 – 25.4	16.4 – 25
19	17.6 – 25.4	16.5 – 25
20 – 60	18.5 – 25	18.5 – 25

For each meal, we can define the domain as follow:

```
fluents:      eaten(T) requires meal(T).
          idr(Z,V,Y,W) requires #int(Z), #int(V), #int(Y), #int(W). %Count calories & initial set
          idrE(E,P,I,C) requires #int(E),#int(P),#int(I),#int(C).     %calories, proteins, fat, carbs
          idrMax(X,R,B,A) requires #int(X),#int(R),#int(B),#int(A). end.
          actions:      eat(F,T) requires meal(T),item(F,T,E,P,I,C).
```



Results

Your Boby Mass Index is: 27.3

More information

Status You should lose some weight to stay healthy =)

Diet Goal

● Gradual weight loss 1/2 lb per week

○ Moderate weight loss 1 lb per week

Your Recommended Daily Intake is: 1854 calories per day

Back Menu

BMI

Recommended values for you:

BMI	Weight
19-25	48.6-64

Ideal weight for reference: 52.4

Weight loss to stay healthy: 6 Kg

Back

Diet

Calories 0%

More information

Breakfast 0

Lunch 0

Dinner 0

Automatic Plan

Exit Menu

Planning

Plan for:

□ BreakFast

□ Dinner

□ Supper

Back Ok

Figure 3a) BMI Figure 3b) Main interface & planning function

In our case, the only action is the user can do is to eat or have food one by one, soup first, main course second and a beverage last. The goal is to optimize the number of carbs (g), proteins (g) and fats (g), that may be between the values for idrE (enough) and idrMax (maximum allowance), and near to the calorie requirement .This range is set by 80% for idrE and 110% for idrMax of the ideal values calculated before. The rules following the declarations of actions and fluents (always) describe the transitions and constraints on the initial states of the domain (for space reasons we do not present all code). Finally, the goal: section defines the goal to be reached and the plan length.

```
always:executable eat(F,T) if not eaten(T),item(F,T,E,P,I,C).
       nonexecutable eat(F,2) if not eaten(1).
       nonexecutable eat(F,3) if not eaten(2).
       forbidden idr(Z,V,Y,W), idrMax(X,R,B,A), Z > X, V > R, Y >  B,W > A.
       caused idr(Z,V,Y,W) after eat(F,T), item(F,T,E,P,I,R), idr(A,B,C,D), +(A,E,Z),
                          +(P,B,V), +(C,I,Y), +(R,D,W).                        caused eaten(T)
after eat(F,T).                                                      caused end after idr(Z,V,Y,W),
idrE(E,P,I,C), Z > E, V > P, Y > I, W > C.                 inertial eaten(T).
                                                   inertial idrE(E,P,I,C).
                         inertial idrMax(X,R,B,A).
       noConcurrency.
```

The user's initial conditions are automatically set by program, but continuing with the example they looks like this for breakfast:

```
initially:      idr(0, 0, 0, 0).   idrE(417,13,12,51).      idrMax(500,17,15,63).
```

Extra conditions were set here if the user decide to eat other food than the suggested for the program (see figure 4a) then user can modify this proposal.

```
       position_target(2,3).                   % target point requested by user via interface
       item(soup1,381,29.25,19.13,22.02).      % foods
       item(soup2,102,3.05,2.93,17.93).        % foods
       …
       item(mainCourse1,443,9.74,32.02,31.88).
       item(mainCourse2,438,13.5,31.37,27.35).
       …
       item(drink2,136,0.26,0.07,35.18).
       item(drink3,2,0.28,0.05,0.09).
       …
```

```
food(F):- item(F,_,_,_,_,_).
meal(1).
meal(2).
meal(3).
```

goal: end?(4) %This rule defines the goal (desired target) to be reached and the plan length.

3.1. Running it in DLV^K

Assume that the above background knowledge, planning program and the initial domain is separated in files dieta.dl and dieta.plan and initial.plan , respectively. The execution of the command:

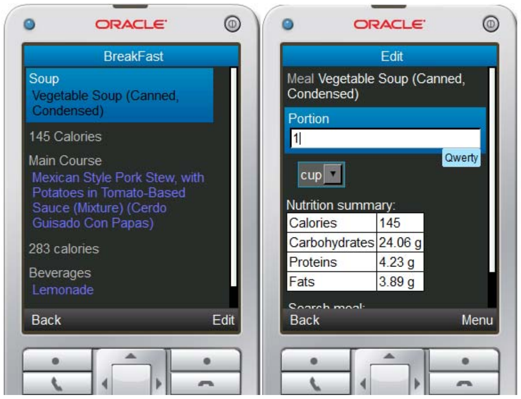
```
# dlv dieta.dl dieta.plan initial.plan -FP -N=1000 %computes the result for breakfast:
```

```
PLAN: eat(soup7,1); eat(mainCourse7,2); eat(drink5,3); (no action)
```

For lunch and dinner respectively:

```
PLAN: eat(soup5,1); eat(mainCourse2,2); eat(drink9,3); (no action)
PLAN: eat(soup10,1); eat(mainCourse8,2); eat(drink2,3); (no action)
```

This plan is most adequate to ideal values proposed (as you can see in Figure 4b). The command-line option –N=1000 tell dlv to compute operations until number 1000 as maximum value. Option –FP is used to invoke the planning front-end in dlv. Alternatively, one can use the options –FPopt and FPsec, which explicitly enforce optimistic and secure planning, respectively. For planning problems with a unique initial state a deterministic domain, these two options do not make a difference, as optimistic and secure plans coincide in this case. This solution is clear, simple and all rules in IR and CR have to satisfy the safety requirement for default negated type literals, i.e., each variable occurring in a default negated type literal has to occur in at least one non negated type literal or dynamic literal.



BreakFast

Soup
Vegetable Soup (Canned, Condensed)
145 Calories

Main Course
Mexican Style Pork Stew, with Potatoes in Tomato-Based Sauce (Mixture) (Cerdo Guisado Con Papas)
283 calories

Beverages
Lemonade

Edit

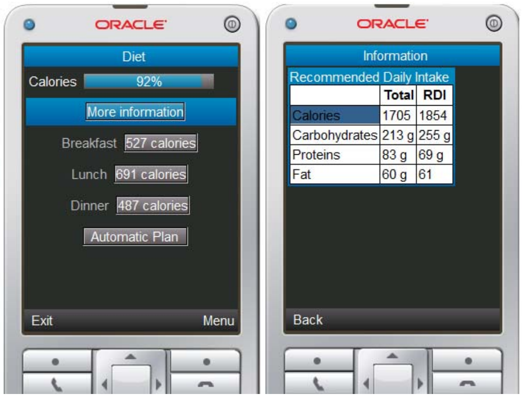
Meal Vegetable Soup (Canned, Condensed)

Portion: 1 cup

Nutrition summary:

Calories	145
Carbohydrates	24.06 g
Proteins	4.23 g
Fats	3.89 g

Figure 4a) Interface to modify diet



Diet

Calories 92%

Breakfast 527 calories
Lunch 691 calories
Dinner 487 calories

Information

Recommended Daily Intake

	Total	RDI
Calories	1705	1854
Carbohydrates	213 g	255 g
Proteins	83 g	69 g
Fat	60 g	61

Figure 4b) Diet Summary

4. Mobile Architecture

New technologies such as: Wireless Application Protocol (WAP), General Packet Radio Service (GPRS) and 3G (3rd generation) further increase communication technologies and access to information. Wireless Application Protocol (commonly referred as WAP) is an open international standard for application layer network communications in a wireless communication environment. Its main use is to enable access to the Mobile Web from a mobile phone or PDA.

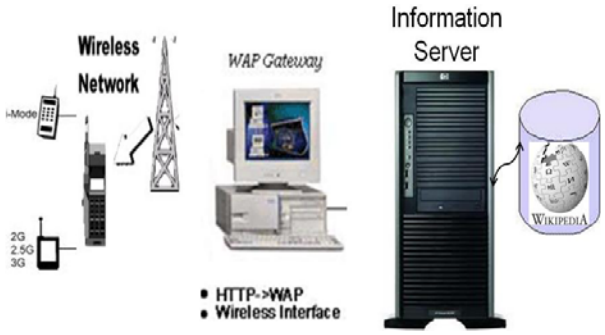


Figure 5 Architecture for our application

The recent advances in mobile technology and wireless basic requirements of the applications of WAP have helped improve trade and mobile services (see figure 5). This protocol is a secure specification allowing users to access services and information instantly through wireless mobile devices [14]. WAP is composed of the following form: uses Wireless Markup Language (WML), which includes the Handheld Device Markup Language (HDML). WML can also trace its roots to eXtensible Markup Language (XML).

The best known markup language is Hypertext Markup Language (HTML). Unlike HTML, WML is considered a Meta language. WAP also allows the use of standard Internet protocols such as UDP, IP and XML. Although WAP supports HTML and XML, the WML language (an XML application) is specifically devised for small screens and one-hand navigation without a keyboard. WAP also supports WML Script. It is similar to JavaScript, but makes minimal demands on memory and CPU power because it does not contain many of the unnecessary functions found in other scripting languages. The WAP programming model is similar to the Web programming model with matching extensions, but it accommodates the characteristics of the wireless environment. In figure 6, shows the proposed architecture for nourishing system. The module for obtain a custom and automatic plan to support people improve their general health was implemented as Web services using standard languages [15]. It takes data from a mobile device; these data are sent to the web service which returns a specific plan obtained from DLV^K system, similar to [10].

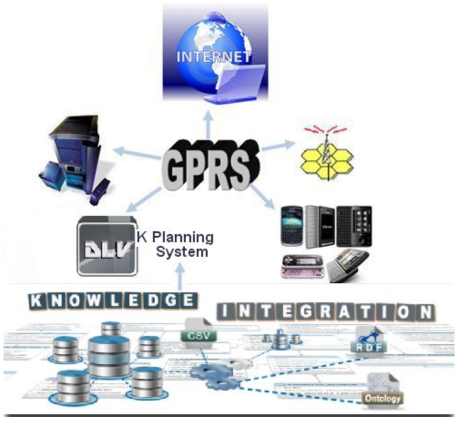


Figure 6 General architecture

The architecture used consists of the following modules:

- Mobile Planning Nourishing (mPN): is the interface of the mobile device to the user, which is responsible for communicating via GPRS to the web services.
- Web service K Planning System: is the web service to meet all the demands of mPN from the site, to send all requests to Knowledge Integration engine to be resolved.
- Knowledge Integration engine: is responsible for find the best diet plan.
- Repository of food: Food repository in which DLV^K seeks for nearest plan to ideal values proposed, and this will be sent to the user.

In a broader way, the user on their mobile device input his/her their information profile, the system provides mobile communication system to the Web Service by GPRS connection (for economy and accessibility by users in case of Mexico 4 cents per KiloByte), in the web service the user's profile is used to compute a reference diet values that is sent to Knowledge Integration Engine [16]. This engine finds the most adequate plan using meals from the Repository of food. The plan is returned to the user by previously established connection to the mobile device where the plan is shown.

5. RESEARCH METHODOLOGY

Research methodology used for the evaluation of our tool was based in discourse analysis to describe the young people's perceptions at the beginning of the experiment and after following the meal plans proposed by our tool.

5.1. Research setting

We selected 52 high school students to participate in the experiment. The selection was done based on tastes and eating habits of students and the ownership of appropriate cellular phone. The plans are made individually by user request considering: Age; Gender; Weigth; Heigth; and Physical activity. These five characteristics allow us to have an overall background about the food that the user must have. Thus, the system provides you to your meal plan allowing capture ingested food to see if it has followed correctly, or otherwise, the system performs a re-planning considering the excess.

Young people used our application for carrying out searches about balanced meals, exploding the mobility, dynamics, availability and accessibility properties of the cellular phones and technologies embedded in them, such as: visualization of food content, suggested meals, average weigths ideals, share recipes via messages, nutritional video sharing, etc. In this paper it has been shown that language K is very expressive in terms of planning and reasoning about actions, allowing to encode even hard planning problems with alternative preconditions of actions, and nondeterministic actions effects. With the current implementation of DLV^K we provide a system which allows a wide variety of domains to be encoded and solved.

5.2. Data collecting means and tool

We used the following means and tools to collect data regarding the participants' progress using mobile phone, tablets and the web application.

Blog: We used a pre-constructed blog in order to suggest ideas regarding the use of mobile phones in the process of a healthy meal planning and to inquire about this use. We inform participants that they could ask questions in the blog, add remarks, add feedback, comment, document events and actions and write about their feelings regarding the experiment. They also asked the users to write about their expectations and the activities that they would like to be engaged with.

The blog started with a welcoming text as follow: If this isn't the first time that losing weight was one of your New Year's resolutions, then maybe you just haven't had the right diet motivation. It turns out that the key to losing and keeping weight off isn't simply a matter of what you eat or how much you exercise - it's your attitude. For this, We informed the students that they could ask questions in the blog, add remarks, add feedback, comment, document events and actions and write about their feelings regarding the experiment.

Social networks: The use of social networks like Facebook, Twitter, MySpace, etc., provided the interaction between the young participants in the experiment. Comments regarding our tool were in the sense that this would enable them to take a strong attitude to lose weight. Even, comments on social networks were more spontaneous and short, which allowed us to complement with the blog.

5.3. Interviews with the users

They interviewed each participant for twenty minutes about her/his experience using our application in mobile phones, tablets an web application. The interviews were semi structured to analyze and characterize the users' perceptions of their progress.

5.4. Data processing and analysis

Our analysis was based on discourse analysis [12] which can be described as an "analysis which studies practices of producing knowledge and meanings in concrete contexts and institutions", and adds: "Discourse analysis systematizes different ways of talking in order to make visible the perspectives and starting points on the basis of which knowledge and meanings are produced in a particular historical moment. It pays attention to the way in which discourses produce and transform social reality, and makes it possible to evaluate the practical consequences of different ways of approaching a particular phenomenon." [13]. we used discourse analysis to analyze what they are eating and to suggest a plan of healthy eating.

In general, the grounded theory approach has three stages:

Open coding – Axial coding – Selective coding.

Open-Coding: Identification of repeated behaviors that can be characterized. The procedure involves dividing each type of the gathered data into segments and examining these segments for similarity and difference. This stage concludes with putting the similar behaviors in the same category and characterizing each category.

Axial coding: Next, we examine the relations among these categories and their subcategories.

Selective coding: After refining the categories, subcategories and their characteristics and relations, we try to identify one or two main or core categories that could be used to connect the rest of the categories with them, and to build a conceptual frame of the studied phenomenon.

5.5. Discussion

This analysis was based on the users' comments in the blog and interviews. Then, we used open and axial coding to categorize the users perceptions of the qualities of the tool and the characteristics of the tool environment when the activity took place in the mobile phone environment. This coding was done based on the users' comments in the blog and in the interviews too. Our users perceive the environment of mobile health using mobile phones as having different characteristics than the traditional one. These characteristics enrich their perception about health and make it more enjoyable. Some users identified herself with the experiment because it makes eat easier, simpler, and nice. It should be noted that we consider the novelty of the experiment to be the main reason for the users' astonishment, because many users mentioned this novelty in the blog.

We can see that the novelty of the experiment is a main factor for the users' participation and involvement. Moreover, the users expected the experiment to be enjoyable and a fun one, and this expectation was realized because of the interesting eating qualities and the mobile phone features. These fun feelings motivated the users making them like the experiment and identify with it, and consequently believe in its success.

The users were also interested in the system's ability to capture what they eat and to make a re-planning because: they enjoy eating with friends away from home; the possibility of bringing the system in his mobile phone with fun encouraged them to eat healthy. From the results we note that the application has considerable acceptance since it is used not only personally, but also can be shared with family, friends or colleagues, do not need any training to use, the need for information users was almost completely satisfied with what they found a good degree of utility to the application. On the other hand, it is necessary to be enriched with more meals and some videos showing the preparation of these. Also, the comments were very rewarding on reducing overweight in 76% of participants. Although this 76% only managed to lose about 800 grams, this is because experiment was only for 2 months and participants neglect their eating

6. CONCLUSIONS

In this paper it has been shown that language K is very expressive in terms of planning and reasoning about actions, allowing to encode even hard planning problems with alternative preconditions of actions, and nondeterministic actions effects. With the current implementation of DLV^K we provide a system which allows a wide variety of domains to be encoded and solved. With the application developed to help the nutritional issue in Mexico, we could create a tool that allows everyone to have access to a custom meal plan without a dietitian. However, there are some limitations or possibilities for improvements that only further research will solve. Some of those improvements would be 5 meals per day, optional snacks and larger food database.

ACKNOWLEDGEMENTS

Thank you very much to Autonomous University of Puebla for their financial support. Also, is important to note the excellent participation of BSc. Madrid in the coding of this novel mobile application.

REFERENCES

[1] V. Lifschitz. Action languages, answer set and planning. The logic programming paradigm – A 25 Year perspective. Springer, 1999.

[2] M. Gelfond and V. Lifschitz. Representing action and change by logic programs. Journal of Logic Programming. Vol. 17, pp. 301-322;1993.

[3] M. Gelfond and |V. Lifschitz. Action Languages, Linkoping electronic articles in computer science and information science, vol. 3, nr 16. 1998.

[4] Drew McDermott. The Planning Domain Definition Language, Technical Report CVC TR-98-003/DCS, Yale University, 1998.

[5] Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A., Planning under incomplete knowledge. CL2000, No. 1861 in LNAI Springer Verlag, London, UK, pp. 807-821, 2000.

[6] Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A., A Logic Programming approach to Knowledge-state Planning: Semantics and Complexity. Tech Rep. INFSYS RR-1843-01-11. Tu-Wien, 2001.

[7] M. Gelfond and |V. Lifschitz. Classical negation in logic programs and disjuctive databases. New generation computing 9, pp. 365-385, 1991.

[8] Eiter T., Leone N., Mateis C., Pfeifer G., Scarcello F. The KR system dlv: progress report, comparasions and benchmarks. In KR-98, Morgan Kaufmann Publishers, pp.106-417, 1998.

[9] Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A., A Logic Programming approach to Knowledge-state Planning, II: The DLVK System. Tech Rep. INFSYS RR-1843-01-12. Tu-Wien, 2003.

[10] Fernando Zacarias F., Rosalba Cuapa C., Guillermo De Ita L., M.A. Balderas, Dionicio Zacarias F. Question Answering:A novel approach. The international journal of Multimedia Technology, USA, 2013.

[11] Harris J, Benedict F. A biometric study of basal metabolism in man. Published by The Carnegie Institute of Washington. Publication No. 279, USA, 1919.

[12] Wetherell, M. & Potter, J. (1988). Discourse analysis and the identification of interpretive repertoires. In Charles Antaki (Ed.), Analysing everyday experience: A casebook of methods (pp. 168-183). London: Sage.

[13] N. Baya'a and W. Daher. Learning Mathematics in an authentic mobile environment: The perceptions of students. doi: 10.3991/ijim v3s.813. 2009

[14] Damian Nowak and Krzysztof Walkowiak. Pre-Paid charging system for Sip-p2p commercial applications. Journal IJCNC Vol. 3 No. 2, march 2011.

[15] Kazuya Odagiri, Naohiro Ishii, Rihito Yaegashi and Masaharu Tadauchi. A portal system and its application based on DACS web service. Journal IJCNC, vol 2. No. 1, 2010.

[16] Nicholas Cooper and Natarajan Meghanathan. Impact of mobility models on multi-path routing in mobile ad hoc networks, Journal IJCNC, Vol. 2, No. 1 2010.

Authors

Prof. Dr. Fernando Zacarias Flores is full time professor in the computer science department into the Benemerita Universidad Autónoma de Puebla. He has directed and participated in research projects and development in this area from 1995, with results that they have been reported in more than 50 publications of international level. And one of the most important is "Answer set programming and applications". Conacyt project, Reference number: 37837-A. He has been a member of several international committees such as: The IEEE Latin America Transaction, IAENG – International journal of Computer Science, International conference on Advances in Mobile Computing and Multimedia, etc.