PLANNING SOLUTIONS IN THE REAL WORLD

Fernando Zacarias[1],Rosalba Cuapa[2], Guillermo De Ita[3], J.C. Acosta[4], Daniel Torres[5]

[1,3,5]Department of Computer Science, Autonomous University of Puebla, Puebla, México
[2]Faculty of Architecture,Autonomous University of Puebla, Puebla, México
[4]Faculty of Engineering, Universidad Autónoma del Estado de México, México

ABSTRACT

In the last years,researchers have made significant progress on robot planning, leading to impressive real-time planners for such challenging tasks as driving, flying, walking, and manipulating objects. In this paper we present a novel architecture to support applications in artificial intelligence in environments fully observable, i.e. in classical planning.On the other hand, we developed ASP-based applications that allowgoing of classical planning to a dynamic planning, according to the real world execution through logic programming where answer sets correspond to solutions, similar to of answer set programming. Furthermore, our systems allow online planning and replanning, depending on whether plan requires revision during execution and replan accordingly.

KEYWORDS

Planning, Wireless Network,WAP Protocol, Answer sets, Logic programming.

1. INTRODUCTION

Planning is an area within artificial intelligence widely studied and mature (AI) for use on real world problems as shown by a number of current applications. Moreover, planning and reasoning about actions and change are relevant fields of AI research since its beginnings. One of the methodologies that have recently been used in planning is declarative approaches based on logic programming. In declarative approaches, the planning problem specification is realised in a logic-based declarative formalism. Find the set of states that define the plan is reduced to elementary problems in computational logic such as satisfiabilitychecking, which are solved by an efficient computational engine as: SAT, smodels and DLV^K(Declarative Action Language *K*).In this paper we presents as action languages based on logic like *C* or *K* allow formalizing complex planning problems. Besides, these languages allow modelling non-determinism and management incomplete knowledge in a clear and flexible manner. Action language *K*is based on logic programming where fluents represent the mental state that agent has about their worldand might be true, false or undefined in each mental state.

With these examples we show that this view of knowledge states can be fruitfully applied to several novel planning domains. In *K*, it is possible to reason about states of knowledge, in which a fluent might be true, false or unknown, and states of the world. In this way, we can deal with uncertainly in the planning real world at a qualitative level, in which default and plausibility principles might come into play when reasoning about the current or next states of the world, the effects of actions, etc. This characteristic and properties allows different approaches to planning, including traditional planning or classical planning and planning with default assumptions or forgetting.

DOI : 10.5121/ijaia.2014.5304

Thus, a novel architecture for mobile applications can be development whose inference engine (DLV^K) is hosted on a server on Internet. The communication between mobile application and server is through WAP protocol via GPRS service. This configuration allows the development of low-cost mobile applications (4 cents for kb) that use large systems like DLV, impossible to stay in a mobile device. Furthermore, we present applications using this architecture that allow showing how we can model real problems in planning applications, in the first instance, the typical problem of finding a route to get from one point to another (within the campus of the Autonomous University of Puebla), and second, a problem of personalized nutritional planning (with replanning at all times and anywhere).

We can define a planning problem as the following task: Given an initial configuration (an initial state), several actions, their preconditions and effects, find a sequence of actions to achieve a state in which a particular goal hold.

When we think of planning, we must focus on the representation of actions and how to model the world, also reason about the effects caused by actions, and define techniques for efficiently searching the space of possible plans. Currently, there are several modelling languages logic-based that allow reasoning about actions and planning of these, such as: [1], [2], [3] and [4]. We have decided to use DLV^K, because this allows us to model formally the effects of actions on the world. It is a new front-end called *K*, developed on DLV (remembering that the language K allows the representation of the state of knowledge) for planning [5] and [6]. DLV^K is flexible, and allows modelling the sequence of states that define a corresponding plan to states of the world (i.e., states of complete knowledge) and reasoning about them as a particular case. *K* is inspiredon answer set semantics [7] rather than in classical logics. Furthermore, it supports the default negation and true negation;these features allow it to fully exploit the answer set semantics to deal with incomplete knowledge.

We say a plan is correct and it solves (solves according to the system sown Internal model of the world) a real planning problem if the initial configuration (initial state) projected through of a set of states called plan, satisfies the problem goal. The plan solves a planning problem if the real-world state that results when the plan is executed from the problems initial state satisfies the problems goal. Next, we present our first application on planning, with a classical problem about planning routes.

2. CLASSICAL PLANNING USING DLV^K

In general, in classical planning, a typical problem is to find a particular building in a huge campus as: universities, industrial complex and big cities, for example, how go from computation department to administrative department? (see figure 1).In this context, it's important that we understand how to model such problems as a set of disjunctive rules and try to plan for innovation at each stage of its life cycle. Thus, in a first instance, we present a novel application developed under the iPad platform that allows us to obtain the desired route to go from one point to another. Next, we show (in figure 2) the interface of the mobile application developed (in apple's iPAD 3 with retina display) for the campus "Universidad Autónoma de Puebla" as in [10], [16]. Next, we define the problem domain as follow:

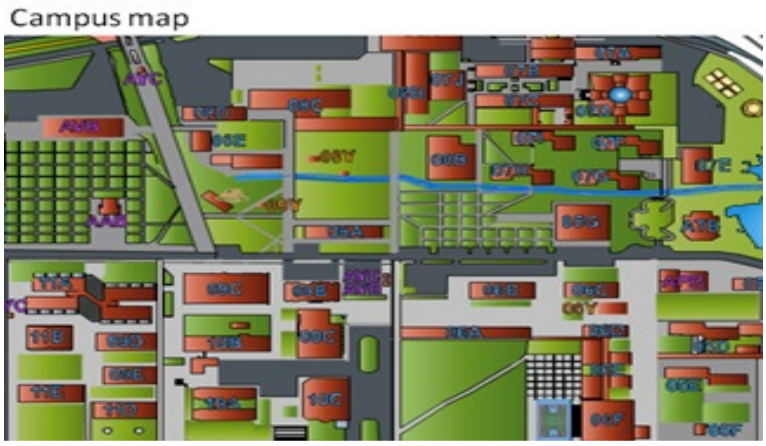

Figure 1. Campus of Autonomous University of Puebla

% We define the agent's world with their basic properties, which can change over time

fluents: targetdown. userlive.
position_building(X1,Y1) requires campus(X1,Y1). position_lake(X4,Y4) requires campus(X4,Y4). position_garden(X5,Y5) requires campus(X5,Y5). position_target(X6,Y6) requires campus(X6,Y6).

start(X2,Y2) requires campus(X2,Y2). user(X3,Y3) requires campus(X3,Y3).

% In DLV^K we can define actions that the agent can execute, each action can change the state

actions: moveRight costs 1. moveLeft costs 1. moveUp costs 1.moveDown costs 1. reachTarget. death.

Where actions may have associated costs and each action has costs 1, resulting in plans, where a minimum number of actions are executed to achieve the desired plan, Also,the user interface givesa route that can be recalculated at any time, anytimes. Next, we have to define the transitions and constraints on the initial states of agent domain. Finally, the goal section defines the goal to be reached and the plan length.

Figure 2. Main system interface

Transitions between different states are defined in clause "**always**". These transitions are atomic changes and they are represented by a previous state, a set of actions and final state representing goal.

always: executable reachtarget if position_target(X,Y), user(X,Y).
executablemoveRight if user(X,_), X<4.
executablemoveUp if user(X,Y), Y<5. caused -targetdown after reachtarget. caused user(X,Y) after reachtarget, user(X,Y). causeduserlive after reachtarget, userlive. caused -userlive after reachtarget, -userlive. caused user(R,Y) after moveRight, user(X,Y), add(X,1,R). causedtargetdown after moveRight, targetdown. caused -targetdown after moveRight, -targetdown. causeduserlive after moveRight, userlive.
caused - userlive after moveRight, - userlive.
caused user(X,R) after moveUp, user(X,Y), add(Y,1,R).
causedtargetdown after moveUp, targetdown. caused -targetdown after moveUp, - targetdown. causeduserlive after moveUp, userlive. caused – userlive after moveUp, - userlive.

On the other hand, we have to define the user's initial position: user(E,E), it is determined byhis geoposition. In a similar form, we have to define gardens, lakes and buildings into campus. These are represented as follow:

position_garden(1,2). position_garden(1,4). targetdown. userlive.
position_lake(3,1). position_lake(3,3).
position_building(2,1).position_building(4,1). position_building (3,2).
position_building(2,3).position_building(4,3). position_building (3,4).
position_garden(2,3).

Finally, we have to define goal to be reached by our agent.

goal: -targetdown, user(E,E), userlive ? (#maxint)

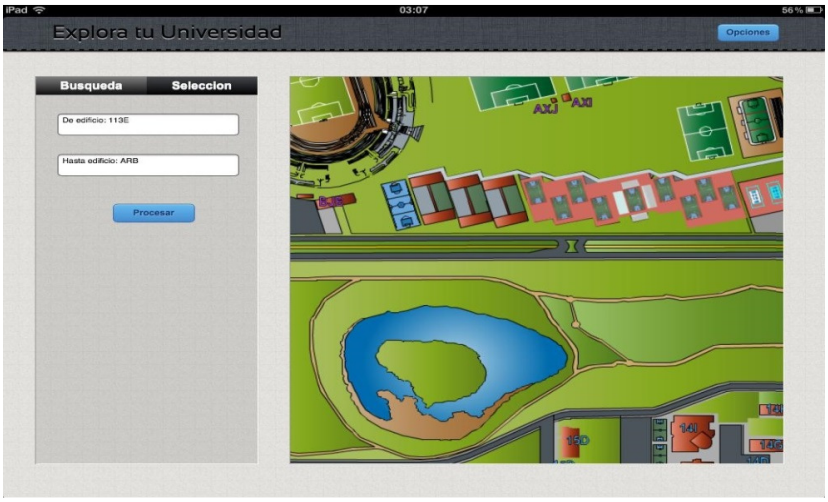

Figure 3. Location of a particular site

2.1. Running the Campus domain in DLV^K

In DLVK modelling program is mainly to two files; in first file"cu.bk" we define the background knowledge needed for our problem, and in second file "cu.plan " we define the planning program, respectively. To execute this program from the command line is as follows:

```
C:\ dlv cu.bkcu.plan –FP –n=1      % Computes the result on server and this is sent to the
                                   % mobile device in no more than 30 segs.
```

PLAN: moveUp; moveUp; moveUp; moveUp; moveRight; moveRight; moveDown; moveDown; moveLeft; reachtarget; COST 9.

DLV^K shows the minimum plan found for our goal (as you can see in figure 3). Each one of movements suggested in the plan states represents the way to achieve the goal. The command-line option –n=1 tell dlv to compute only one plan, optimal. Option –FP is used to invoke the planning front-end called action language K embedded in dlv. Furthermore, DLV support DLV supports switches that allow finding a safe and optimistic plan. 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.

In figure 4 is presented in graphical form the path found by our system, this solution the solution is shown to scale and shown on the iPad so that the user can follow it without any problem besides, this 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.

Figure 4. Map showing the route to reach the goal

3. EXTENDING CLASSICAL PLANNING TO REAL-WORLD

In this section, we present a solution to one of the problems currently faced by many countries in the world, the problem of obesity. Almost all countries are experiencing an obesity epidemic. In neighboring countries like United States of North America, obesity is becoming ever more common among younger adults and children. In Mexico, most of the child population have poor diets that may have an injurious impact on health. Thus, we have a need to provide a safe, easy-

to-access dietary advice for everyone, in special way for Mexican children. Therefore, in this paper a novel mobile application is deployed, this application allows us to obtain an automatic plan to support people improve their general health.

Figure 5.Personal data to determine the plan

In figure 5 is shown the interface of the mobile application developed to capture the necessary information about the user before computing any plan, this information correspond to: age, weight, height and physical activity. With this information we can to calculate the Body mass index is defined as the individual's body mass divided by the square of their height (BMI, for its acronym in English). Next, we can verify that the user is within the normal range according to the Table 1; and estimate the basal energy expenditure (BEE, for its acronym in English) using the Harris-Benedict equations [11]. The BEE is calculated by multiplying the value assigned to your physical activity,whose values can be 1 or 2. With this we can calculate the calorie requirement per day (kcal/day) as you can see in the example shown in figure 6.

Immediately, our system has the necessary information to find the right plan that allows user to reach the calorie requirement per day; 55% carbohydrates, 15% proteins, and 30% fats divided into 3 meals(i.e., at breakfast,lunch and dinner). We note that if any of these meals is changed by the user, then our system can replan at anytime (as you can see in figure 6).

Figure 6.Results of: BMI, Diet, planning and replan

We can define the domain for each meal 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).

Table 1. 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

In our case, the user can eat each meal components, ie, soup, main course and drink. Recall that the main goal of our system is to monitor the user does not exceed the number required by day: 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 defined as follow: 80% for idrE and 110% for idrMax of the ideal values calculated before. Next, we must define the rules that describe the transitions and constraints on the initial states of the domain. Lastly, we set the goal to achieve: thissection 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).
forbiddenidr(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). inertialidrE(E,P,I,C). inertialidrMax(X,R,B,A).

.

The user's initial conditions are automatically set by program, and 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).

At this point, the system can establish additional conditions if the user decide to eat other food than the suggested for the program (as you can see in figure 7, modify diet) then user can modify this proposal and our system seek to modify the suggested diet.

```
position_target(2,3).                    % target point requested by user via interface
item(soup1,381,29.25,19.13,22.02).       % foods available
item(soup2,102,3.05,2.93,17.93).         % foods available
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) %In this section we define desired target to be reached and the plan length.

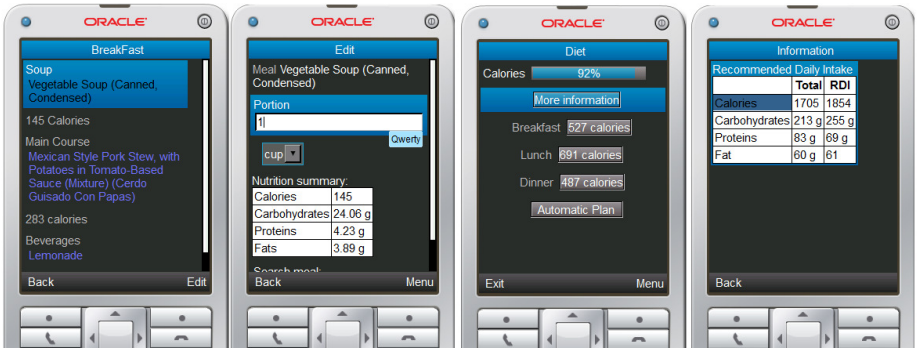

Figure 7. Interface that shows diet and diet summary

3.1. Executing it in DLV^K

The execution of the program will calculate the dietary plan suggested is contained in the following files: “dieta.dl” contains the basic knowledge about the user, “dieta.plan” contains the rules for calculating the dietary plan, and “initial.plan” contains the initial configuration, as well as the objective to be achieved. The execution of the command is:

```
C:\ dlv dieta.dldieta.planinitial.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 (Figure 7, diet summary). The command-line option –N=1000 tell dlv to compute operations until number 1000 as maximum value. Switch –FP invokes the front-end called action language K. For those planning problems which have only a single initial state a deterministic domain, is not necessary to include these two switches. The solution obtained is clear, simple and all rules in IR and CR have to satisfy the safety requirement for default negated type literals.

4. MOBILE ARCHITECTURE

Actually, there are new wireless technologies such as: the international wireless standard called "Wireless Application Protocol" (WAP), some services offered by WAP protocol are: short message service, data service switching circuit and by packet-switched, in particular, we use GPRS as means of data communication. Furthermore, 3G increase communication technologies and access to information. WAP is an open international standard for application layer network communications in a wireless communication environment. In our application we use wap as means by which enable access to the Mobile Web from a mobile phone or PDA.

The recent advances in mobile technology and wireless basic requirements of the applications of WAP have helped improve trade and mobile services, allowing novel architectures as proposed in this paper (see figure 8). This protocol is a secure specification allowing users to access services and information instantly through wireless mobile devices [10],[12], [16]. The WAP protocol is comprised of the following layers: uses Wireless Markup Language (WML), included in WAP version 1.x, it includes the Handheld Device Markup Language (HDML). WML cut users off from the conventional html web, however it can also trace its roots to eXtensible Markup Language (XML).

Figure 8. Novel architecture for real applications

Until now, the most popular language is html(Hypertext Markup Language). While html is a basic language and solid, WML is considered a Meta language. On the other hand, WAP allows the use of standard Internet protocols, within which stand: UDP, IP and XML. Although the WAP protocol supports languages such as html and XML, this has its own language called WML (an XML application) is specifically devised for small screens and one-hand navigation without a keyboard. Also, WAP supports WML Script, the primary language of the WAE. It is similar to JavaScript, although their memory requirements and CPU are minimal because it does not contain many of the unnecessary functions found in other scripting languages. The WAP programming model follows the style of web programming with matching extensions, although in the case of WAP, this conforms to the characteristics of its wireless environment.One of the main modules of our application is the one that allows obtain a custom and automatic plan to support user improve their general health, it was implemented as Web services using standard languages [10], [13]. It mobile device send the request through the service GPRS; these data are received by web application which returns a specific plan obtained from DLV^{K} system, similar to [10].
Our proposed architecture consists of the following modules:

- Mobile Planning Nourishing (mPN): This is the mobile application that allows user to calculate the suitable plan for your health, 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: It is developed in the K action language 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.

Thus, through our mobile application the user can register their profile at all times. Next, the system provides mobile communication system to the Web Service by GPRS connection, it response time does not exceed 20 seconds and besides it is economic and easy access by users (in case of Mexico, 4 cents per KByte), in the web service the user's profile is used to compute a reference diet values that is sent to Knowledge Integration Engine, this in order to keep a record of their behavior. This engine finds the most adequate plan using meals from the repository of food. Immediately the plan is returned to the user through GPRS, previously established connection to the mobile device where the plan is shown. Is important to note that this plan is suggested by our system, however, the user has the last word.

5. RESEARCH METHODOLOGY

Research methodology used for the evaluation of our application 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 voluntarily and enthusiastically. The selection was done based on tastes and eating habits of students and the ownership of appropriate cellular phone or mobile device. The plans are made considering a personalized way and to user´s request considering following five parameters: Age; Gender; Weigth; Heigth; and Physical activity. Based on these five features, our system can have an overall background about the food that the user must have. Furthermore, our system allow capture ingested food to see if it has followed correctly plan, and if necessary, the system performs a re-planning considering the differences between actual consumption and suggested.

Young people used our application for carrying out searches about balanced meals, even looking meals that awaken their appetite (according to feedback gathered from interviews), 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: One of the most interesting media and that provide surprising comments was a pre-constructed blog whose task was to collect ideas regarding the use of mobile phones in the

process of a healthy meal planning and to inquire about this use. We commented to all 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. Suggested them to comment on the particular way that the system performs replanning. Also,we recommend users to write about their expectations and the activities that they would like to be engaged with.
To begin participation in the blog, put a welcome text as follow: We hope to support you in your goals new year, weight loss, for this, we invite you to use our application of dietary planning. 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 reason, our application will not only help to keep that positive attitude about their nutrition, but also, will allow you to rebuild your dietary plan whenever necessary to correct your plan.

Social networks: The use of social networks like Twitter, MySpace, Facebook, etc., provided the interaction between the young participants in the experiment and allows us to know directly what our participants perceive about our application. Comments regarding our application 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. However, we note that for the analysis of these comments turn to the interpretation of this peculiar form of writing in these social media, since writing is abbreviated and cut and even symbols are used.

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

The data analysis was done under the discourse analysis methodology [14] which is described as an “analysis which studies practices of producing knowledge and meanings in concrete contexts and institutions", moreover, this also incorporates: Discourse analysis regulates or directs the different ways of speaking whose objective is highlight the perspectives as well as the starting points on the basis of which knowledge and meanings are produced in a particular historical moment. Discourse analysis focuses on how discourses are produced, since these transform social reality, and makes it possible to evaluate the practical consequences of different ways of approaching a particular phenomenon [15]. We used discourse analysis to analyze what they are eating to focus on two key points: First, which of the three meals the least performed as is suggested; second, make more rigid the other two meals in order to achieve the main objective.
In addition, our grounded theory approach consists of three stages:

Open coding – Axial coding – Selective coding.

Open-Coding: Identify those behaviors that are repeated more so that we can characterize them.The procedure is to identify and classify the data to form the corresponding classes and examining these classes for similarity and difference. This process or stage concludes with putting the similar behaviors in the same category (or class) and characterizing each category.

Axial coding: Next, should examine the relationships between the different categories and their subcategories.

Selective coding: Finally, refining the categories, subcategories and their characteristics and relations, the primary objective is to identify one or two main 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, social networks and interviews. Thus, 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 same process is performed with the web environment. This coding was done based on the users' comments in the blog, social networks and in the interviews too. Our users perceive the environment of mobile health using mobile phones, tablets and gadgets 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 and ease of the experiment to be the main reason for the users' astonishment, because many users mentioned this novelty in the blog and interviews.

We can see that the novelty of the experiment is a main factor for the users' participation and involvement. Furthermore, the use of tools that are part of everyday life of young people, such as mobile devices, social networks and blogs made this experience a pleasurable activity. Moreover, this activity was realized because of the interesting eating qualities and the mobile phone features. Finally, users felt comfortable and attracted by this experience, and consequently believe in its success.

Another good experience was that 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 can development systemsin a wide variety of domains to be encoded and solved.
With the application developed for our campus and particularly nutritional planning application, we could create a tool that allows users develop a nutritional plan that can be replan at any time to adjust to the food eaten really. This is done when the user registers the food consumed during a meal and these exceed amounts set nutritional plan. This way, our system takes care in all moment that the plan originally set is reached.

On the other hand, our first application that determines the route to get from one point to another is an example of classical planning. With this, visitors to our campus can go to any point in a simple way. However, also some limitations and possibilities for improvements and further research have developed throughout our work.

ACKNOWLEDGEMENTS

Thank you very much to the Autonomous University of Puebla for their financial support. This work was supported by thematic research network PROMEP called "Combinatorial Algorithms and Pattern Recognition".

REFERENCES

[1] V. Lifschitz. Action languages, answer set and planning. The logic programming paradigm - A 25Year 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: Semantics and Complexity. Tech Rep. INFSYS RR-1843-01-11. Tu-Wien, 2001.
[10] Fernando Zacarias F., Rosalba Cuapa C., Guillermo De Ita L., M.A. Balderas, Dionicio Zacarias F. QuestionAnswering: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] Damian Nowak and Krzysztof Walkowiak.Pre-Paid charging system for Sip-p2p commercial applications. Journal IJCNC Vol. 3 No. 2, march 2011.
[13] Kazuya Odagiri, Naohiro Ishii, RihitoYaegashi and MasaharuTadauchi. A portal system and its application based on DACS web service. Journal IJCNC, vol 2.No. 1, 2010.
[14] 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.
[15] N. Baya'a and W. Daher.Learning Mathematics in an authentic mobile environment: The perceptions of students.doi: 10.3991/ijim v3s.813. 2009.
[16] Fernando Zacarías, Rosalba Cuapa, Antonio Sánchez and Iris Cerecedo. Puebla in the palm of your hands.MoMM '11 Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia.pp.260-263,ACM New York, USA @2011.