Analytical Model for Constructing Deliberative Agents

M. Glez-Bedia and J. M. Corchado
Departamento de Informática y Automática
University of Salamanca
Plaza de la Merced s/n, 37008,
Salamanca, Spain

E. S. Corchado and C. Fyfe
Computing and Information System Dept.
University of Paisley
High Street, Paisley,
U.K.

Abstract: This paper introduces a robust mathematical formalism for the definition of deliberative agents implemented using a case-based reasoning system. The concept behind deliberative agents is introduced and the case-based reasoning model is described using this analytical formalism. Variational calculus is used during the reasoning process to identify the problem solution. The agent may use variational calculus to generate plans and modify them at execution time, so they can react to environmental changes in real time. Reflecting the continuous development in the tourism industry as it adapts to new technology, the paper includes the formalisation of an agent developed to assist potential tourists in the organisation of their holidays and to enable them to modify their schedules on the move using wireless communication systems.

Keywords: Case-based Reasoning systems, Deliberative agents, Variational Calculus

1. Introduction

Technological evolution in today's world is fast and constant. Successful systems should have the capacity to adapt to it and should be provided with mechanisms that allow them to decide what to do according to their objectives. Such systems are known as autonomous or intelligent agents [21]. This paper shows how a deliberative agent with a BDI (Belief, Desire and Intention) architecture can use a case-based reasoning (CBR) system to generate its plans. A robust analytical notation is introduced to facilitate the definition and integration of BDI agents with CBR systems. The paper also shows how variational calculus can be used to automate the planning and replanning process of such agents at execution time.

Agents should be autonomous, reactive, pro-active, sociable and have learning capacity. They must be able to respond to events that take place in their environment, take the initiative according to their goals, interact with other agents (even human) and use past experiences to achieve current goals. There are different types of agents and they can be classified in different ways [21]. One type, the so-called deliberative agent with BDI - Belief, Desire and Intention - architecture, uses the three attitudes in order to make decisions on what to do and how to achieve it [10, 11, 21]: their *beliefs* represent their information state - what the agents know about themselves and their environment; their *desires* are their motivation state - what they are trying to achieve; and the *intentions* represent the agents' deliberative state. Intentions are sequences (ordered sets) of beliefs (also identified as plans). These mental attitudes determine the agent's behaviour and are critical if a proper performance is to be produced when information about a problem is scarce [3, 12]. BDI architecture has the advantage that it is intuitive - it is relatively easy to recognise the process of decision-making and how to perform it. Moreover, it is easy to understand the notions of belief, desires and intentions. On the other hand, its main drawback lies in determining a mechanism, which will allow its effective implementation. The formalisation and implementation of BDI agents constitutes the research of many scientists [5, 11, 18]. Some of these researchers criticise the necessity of studying multi-modal logic for the formalisation and construction of such agents, because they haven't been completely axiomatised and they aren't computationally efficient. Rao and Georgeff [17] assert that the problem lies in the great difference between the powerful logic of BDI systems and that required by practical systems. Another problem is that these types of agents don't have learning capability - a necessary element for them since they have to be constantly adding, modifying or eliminating beliefs, desires and intentions.

This paper presents a robust analytical formalism for the definition of computationally efficient agents, which solves the first of the previously mentioned problems. This paper also shows how a BDI agent implemented using a case-based reasoning (CBR) system can substantially solve the problems related to the learning capability of the agents. Implementing agents in the form of CBR systems facilitates their learning and adaptation. If the proper correspondence between the three mental attitudes of the BDI agents and the information that a case-based reasoning system manipulates can be established, an agent will be created not only with beliefs, desires and intentions but also with learning capacity.

Although the relationship between agents and CBR systems have been investigated by other researchers [15, 20, 16], we propose a robust mathematical formalism that will facilitate the efficient implementation of an agent in the form of a CBR system. Variational calculus is introduced to automate the reasoning cycle of the BDI agents; it is used during the reuse stage of the CBR cycle to guarantee efficient planning and re-planning at execution time. Although different types of planning mechanisms can be found in the literature [4,13], none of them allow re-planning at execution time, even though agents inhabit changing environments in which re-planning at execution time is necessary if goals are to be achieved successfully in real-time. Some of the approaches developed use planning techniques to select the appropriate solution to a given problem but without mechanisms to deal with changes in the environment. For instance, in [13,14] a kind of plan schema is introduced that needs to be reprogrammed over time, when the planning domain changes. In [4] an architecture is proposed that tries to be more flexible by using planning strategies to create the plans. If new information from the environment must be introduced into the system, it is only necessary to change the planning domain instead of reprogramming the plan schema by hand. This architecture allows plans to be built that contain steps with no detailed information. Now, in order to establish that the abstract proposed plan is adequate, it is necessary to put it into practice in a real domain. This operation requires a high amount of computational time and resources which may be a disadvantage in, for example, web-related problems. The flexibility of this approach increases the time spent in applying the abstract solution to the real problem, which is a handicap for real time systems.

In this paper, a solution is proposed that deals adequately with environmental real-time problem changes without applying a reprogramming strategy and without the disadvantages shown in [4] because the technique used can solve a planning problem at execution time. This is achieved by using variational calculus during the retrieval stage of the CBR life cycle.

To begin with, the paper will review the concepts of CBR systems and deliberative agents using an analytical notation. Then it will be shown how a CBR system is used to operate the mental attitudes of a deliberative BDI agent. This section also shows the relationship between BDI agents and CBR systems. Then variational calculus will be introduced, and it will be shown how it can be used to define agents with the afore-mentioned characteristics. Finally, together with the conclusions it is shown how it is possible to define an agent for the e-tourism domain using the methodology presented.

2. Case-based Reasoning Systems

Case-based reasoning is used to solve new problems by adapting solutions that were used to solve similar previous problems [6]. The operation of a CBR system involves the adaptation of old solutions to match new experiences, using past cases to explain new situations, using previous experience to formulate new solutions, or reasoning from precedents to interpret a similar situation.

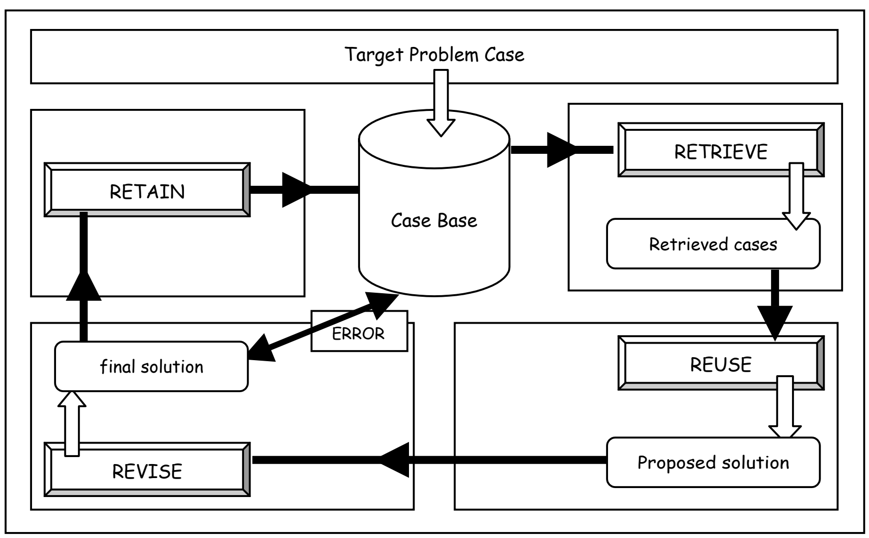

Figure 1: CBR Cycle of Life.

Figure 1 shows the reasoning cycle of a typical CBR system that includes four steps that are cyclically carried out in a sequenced way: retrieve, reuse, revise, and retain [1, 19]. During the retrieval phase, those cases that are most similar to the problem case are recovered from the case-base. The recovered cases are adapted to generate a possible solution during the reuse stage. The solution is then reviewed and, if appropriate, a new case is created and stored during the retention stage, within the memory. Therefore CBR systems update (with every retention step) their case-bases and consequently evolve with their environment.

Each of the reasoning steps of a CBR system can be automated, which implies that the whole reasoning process could be automated to a certain extent [6, 9]. This assumption has led us to the hypothesis that agents implemented using CBR systems could be able to reason autonomously and therefore to adapt themselves to environmental changes. Agents may then use the reasoning cycle of CBR systems to generate their plans.

Based on the automation capabilities of CBR systems we have established a relationship between cases, the CBR life cycle, and the mental attitudes of the BDI agents. Based on this idea, a model is presented that facilitates the implementation of the BDI agents using the reasoning cycle of a CBR system.

3. Implementing Deliberative Agents using CBR Systems

This section identifies the relationships that can be established between BDI agents and CBR systems, and shows how an agent can reason with the help of a case-based reasoning system. The formalisation presented in this paper takes elements of other systems and adapts them to the model presented here. Our proposal attempts to define a direct mapping between the agents and the reasoning model, paying special attention to two characteristics: (i) the mapping between the agents and the reasoning model should allow a direct implementation of the agent and (ii) the final agents should be capable of learning and adapting to environmental changes. An analytical notation has been introduced to facilitate an efficient integration between the BDI agent and CBR system and that allows the use of variational calculus for planning and replanning in execution time. The notation used in the referenced works [14, 16, 20] do not have the required degree of expressivity and complexity to introduce differential calculus tools.

3.1 BDI Agents

The notation and the relationship between the components that characterise a BDI agent are first introduced:

Let Θ be the set that describes the agent environment. If $T(\Theta)$ is the set of attributes $\{\tau 1, \tau 2, \ldots, \tau n\}$ in which the world's beliefs are expressed, then we define **a belief on Θ, that is denoted "e",** as an m-tuple of some attributes of $T(\Theta)$ denoted by $\mathbf{e = (\tau 1, \tau 2, \ldots, \tau m)}$ with $m \leq n$

We call set of beliefs on Θ and denote $\zeta(\Theta)$ to the set:

$\zeta(\Theta) = \{ e=(\tau 1, \tau 2, \ldots, \tau j) /$ where $j = (1,2,\ldots, m \leq n) \}$

Example: It is supposed that the world $T(\Theta)$ includes all the attributes that are needed to characterise beliefs associated to a tourist schedule in the city of Salamanca (European City of Culture, 2002).

$T(\Theta)=\{\tau_1$=monument name, τ_2= visiting time, τ_3= cost,..., $\tau_n\}$

In this world, a belief, for example, *monument*, is represented by a m-tuple of attributes of $T(\Theta)$ that characterise the monument.

Monument =(τ_1=index, τ_2= visiting time, τ_3= cost, τ_4= evaluation)

A particular belief, for example, *the old cathedral* (OC), can be represented by:

Old cathedral=(index=oc, visiting time$\in$(10:00 h., 18:00 h.), cost=3 €, evaluation =2)

Where **OC** (Old Cathedral) is the indexing abbreviation used to store the belief, **visiting time** indicates that Old Cathedral may be visited from 10:00.to 18:00 h., the **cost** to enter the monuments is 3 Euro and **evaluation**, equal to 2, is a subjective index of the quality of the place, defined by an experienced tourist guide, based on the perceptions of visitors. **Evaluation** is a real number between 0 and 3.

We introduce the operator "Λ of accessibility" between m beliefs ($e_1,e_2,e_3,\ldots,e_m$), where we denote: $\Lambda(e_1, e_2, e_3,\ldots,e_m) = (e_1 \wedge e_2 \wedge \ldots \wedge e_m)$ that indicates that exists compatibility among the set of beliefs ($e_1, e_2, e_3,\ldots, e_m$). If any of the belief ($e_1, e_2, e_3,\ldots, e_m$) is not accessible,or if there exists a contradiction,it will be denoted by:$\Lambda(e_1,e_2, e_3,\ldots,e_m) = \emptyset$.

Example: It is 12:00h. p.m. and the agent believes M1, M2 and D(A,A) - which are described in Table 1 - where M1 and M2 are monuments that may be visited and D(A,A) represents the travel from one monument to the other. Both monuments M1 and M2 are in the area A, and going from one to the other by taxi costs 12 Euros. With these believes and given that it is 12:00 o'clock, it is impossible to visit M1 and M2, and therefore the path (M1 ^ D(A,A) ^ M2) can not be constructed and Λ (M1,D(A,A), M2)= Ø.

Table 1. Values of beliefs M1, M2, M3 and D(A,A).

Attribute	Value
Entity	M1
Class	monument
Visiting Time	10-13 hrs.
Visiting Cost	6 €
Time for a visit	1 hr.
Zone or place	A

Attribute	Value
Entity	M2
Class	monument
Visiting Time	10-13 hrs.
Visiting Cost	6 €
Time for a visit	1 hr.
Zone or place	A

Attribute	Value
Entity	M3
Class	monument
Visiting Time	10-14 hrs.
Visiting Cost	6 €
Time for a visit	1 hr.
Zone or place	A

Attribute	value
Entity	D(A,A)
Class	travel by taxi
Time	1 hr.
Cost	12 €

If M2 is substituted by M3 (see Table 1) then (M1 $\wedge$ D(A,A) $\wedge$ M3) is possible, and Λ (M1,D(A,A),M3)$\neq$ Ø, which means that the agent has identified that we can visit the monument M1 and M3, taking into consideration that the time to go from the first to the second monument is given by D(A,A).

Moreover, an intention i on Θ is defined as an s-tuple of compatible beliefs,
$i = (e_1, e_2,\ldots,e_s)$ with $s\in$ IN and $\Lambda(e_i, e_j)\neq 0$
Then, we call set of **intentions** on Θ and denote $I(\Theta)$
$I(\Theta)= \{ (e_1, e_2, \ldots, e_k)$ where $k\in$ IN $\}$

Now a set of parameters will be associated to the space $I(\Theta)$ that characterises the elements of that set. The set of necessary and sufficient variables to describe the system may be obtained experimentally. We call canonical variables of a set $I(\Theta)$ any set of linearly independent parameters $\aleph =(A_1, A_2,..., A_v)$ that characterise the elements $i \in I(\Theta)$.

Example: If the agent identifies a visiting route through the number of monuments to visit (N) and a maximum associated cost (C), then we express it as $\aleph =(A_1,A_2)=(N, C)$. In this coordenates system the following intention:
$i_1= M1 \wedge D(A,A) \wedge M2 \wedge D(A,A) \wedge R1 \wedge D(A,B) \wedge M3 \wedge D(B,B) \wedge R2$

is represented in Table 2.

It also has the values for P (number of monuments visited) and C (total cost of the tour) indicated. Again M1, M2, M3 are monuments, R1 and R2 are restaurants and D(A,A), D(A,B) and D(B,B) represent the journeys between the visited places.

Table 2. **Values of the believes that constitute intention i_1 and values for (P,C) associated.**

Schedule(hr)	10-11	11-12	12-13	13-14	14-16	16-18	18-20	20-21	21-22
intention	M1	D(A,A)	M2	D(A,A)	R1	D(A,B)	M3	D(B,B)	R2
Costs(€)	6	0	6	0	12	0	0	0	12
Time (hr)	1	1	1	1	2	2	2	1	1
Evaluation	1	--	2	--	1	--	2	--	2

attributes	
N	3
C(€)	36

In the same way, a **desire** d on Θ is defined as a mapping between

$$d : I(\Theta) \longrightarrow \Omega(\aleph)$$

$$i = (e_1 \wedge \ldots \wedge e_r) \rightarrow F(A_1, A_2, \ldots, A_v)$$

where $\Omega(\aleph)$ is the set of mappings on $\aleph$.

A desire d may be achieved constructing an intention i using some of the available beliefs, whose output could be evaluated in terms of the desired goals. We denote D(Θ) the set of desires on Θ:

$D(\Theta)=\{d: I(\Theta) \rightarrow \Omega(\aleph)$ / with I(Θ) set of intentions and $\Omega(\aleph)$ set of mappings on $\aleph\}$

Example: The desire function "I want to visit at least three monuments and spend less than 50€", may be expressed as:

$$F(A_1, A_2) = F(N, C) = \begin{cases} N \geq 3 \\ C \leq 50 \end{cases} \text{with} \begin{matrix} N \in (0,10) \\ C \in (0,100) \end{matrix} \Bigg\}$$

Now, after presenting our definition of the agent's beliefs, desires and intentions, section 3.2 defines the proposed analytical formalism for the CBR system.

3.2 Analytical formalism for Case-based Reasoning systems

The necessary notation to characterise a CBR system is introduced as follows. Let us consider a problem P, for which it is desired to obtain the solution S(P). The goal of a case-based reasoning system is to associate a solution S(P) to a new problem P, by reusing the solution S(P′) of a memorised problem P′.

P is denoted as $P=(S_i, \{\theta_j\}, S_f)$ with S_i=initial state, S_f=final state and j=(1,…,m). S(P) is defined as $S(P)=\{S_1, \theta_1, S_2, \theta_2, \ldots, \theta_n, S_{n+1}\}=\{S_k, \theta_h\}$ where k=(1,..,n+1) and h=(1,..,n ≤ m) , $S_1=S_i$ and $S_{n+1}=S_f$,

The state S_k and the operator θ_j are defined as:

$$S_k = \begin{pmatrix} \{O_r\}_{r=1,\ldots,p} \\ \{R_s\}_{s=1,\ldots,q} \end{pmatrix} \qquad \theta_j : S_k = \begin{pmatrix} \{O_r\} \\ \{R_s\} \end{pmatrix} \longrightarrow \theta_j(S_k) = \begin{pmatrix} \{O'_r\} \\ \{R'_s\} \end{pmatrix}$$

where $\{O_r\}_{r=1,\ldots,p}$ and $\{R_s\}_{s=1,\ldots,q}$ are coordinates in which a state S_k is expressed

The coordinates type $\{O_r\}_{r=1,\dots,p}$ are introduced to express the objectives achieved. The coordinates type $\{R_s\}_{s=1,\dots,q}$ are introduced to express the resources used.

Through these definitions, the parameter effectiveness, $\Im$, between two states S and S' can be defined, as a vector $\Im(S, S') = (\Im_x, \Im_y)$ which takes the form

$$\Im_x = \frac{O_r(S') - O_r(S)}{O_r \max} \qquad \Im_y = \frac{R_s(S) - R_s(S')}{R_s \max}$$

The definition implies that ($0 \le \Im_x \le 1$) and ($0 \le \Im_y \le 1$). In particular, if $S=S_i$ and $S'=S_f$, it is denoted $\Im(S_i, S_f)=\Im[S(P)]$ and we call it "effectiveness of a solution". In order to evaluate the rate of objectives achieved and resources used, between S and S', it is necessary to normalise every component of $\{O_r\}_{r=(1,\dots,p)}$, $\{R_s\}_{s=(1,\dots,q)}$.

Then the expressions that have been defined to sum different objectives are:

If $\{O_r(S)\} = (O_1, O_2, \dots, O_P)$ and $\{O_r(S')\} = (O'_1, O'_2, \dots, O'_P)$

$$\Im_x = \frac{\sqrt{\left(\frac{O'_1 - O_1}{\max O_1}\right)^2 + \left(\frac{O'_2 - O_2}{\max O_2}\right)^2 + \dots + \left(\frac{O'_p - O_p}{\max O_p}\right)^2}}{\sqrt{\left(\frac{\max O_1}{\max O_1}\right)^2 + \left(\frac{\max O_2}{\max O_2}\right)^2 + \dots + \left(\frac{\max O_p}{\max O_p}\right)^2}} = \frac{\sqrt{\left(\frac{O'_1 - O_1}{\max O_1}\right)^2 + \left(\frac{O'_2 - O_2}{\max O_2}\right)^2 + \dots + \left(\frac{O'_p - O_p}{\max O_p}\right)^2}}{\sqrt{p}}$$

As $\{R_s(S)\} = (R_1, R_2, \dots, R_q)$ and $\{R_s(S')\} = (R'_1, R'_2, \dots, R'_q)$ it is defined

$$\Im_y = \frac{\sqrt{\left(\frac{R_1 - R'_1}{\max R_1}\right)^2 + \left(\frac{R_2 - R'_2}{\max R_2}\right)^2 + \dots + \left(\frac{R_q - R'_q}{\max R_q}\right)^2}}{\sqrt{\left(\frac{\max R_1}{\max R_1}\right)^2 + \left(\frac{\max R_2}{\max R_2}\right)^2 + \dots + \left(\frac{\max R_q}{\max R_q}\right)^2}} = \frac{\sqrt{\left(\frac{R_1 - R'_1}{\max R_1}\right)^2 + \left(\frac{R_2 - R'_2}{\max R_2}\right)^2 + \dots + \left(\frac{R_q - R'_q}{\max R_q}\right)^2}}{\sqrt{q}}$$

A new parameter is also introduced - efficiency - that measures how many resources are needed to achieve an objective. Given a target problem P, and a solution S(P), we define $\varsigma[S(P)] = \Im_x / \Im_y$, as the efficiency of the solution S(P). The definition implies that $\varsigma(S,S') \in (0,\infty)$.

The meaning of this new parameter is explained later. In this domain, a case C is a 3-tuple *{P, S(P), ℑ[S(P)]}* where P is a problem description, S(P) the solution of P and $\Im[S(P)]$ the effectiveness parameter of the solution, and a CBR´s case base CB, denoted as: *CB={C_k / k=(1,...,q) and* $q \in$ IR} that is a finite set of cases memorised by the system.

3.3 Integration of the CBR system within the BDI Agent

The relationship between CBR systems and BDI agents can be established, associating the beliefs, desires and intentions with cases. Using this relationship we can implement agents (conceptual level) using CBR systems (implementation level). So once the beliefs, desires and intentions of an agent are identified, they can be mapped onto a CBR system. First, a mapping is introduced that associates an index to a given case C_k.

idx:CB→ I(CB)

C → idx(C)= idx{P, S(P), ℑ[S(P)]}= **{** idx **(**S_i**)**, idx **(**S_f**)** **}**=

= **{** [S_i =(O_1,a_1), (O_2,a_2),...,(O_p, a_p), (R_1,b_1), (R_2,b_2),...., (R_q, b_q)],

[S_f=(O'_1,c_1),(O'_2,c_2),..,(O'_p, c_p),(R'_1,d_1),(R'_2,d_2),..,(R'_q,d_q)] **}**

with Oj, $R_k \in$ T(CB) ,a_i, b_j, c_k, $d_l \in$ IR and p, q $\in$ IN

where the set I(CB) is the set of indices of a case base CB that is represented by frames composed of conjunction of attributes of T(CB) and values of the domain.

The abstraction realized through the indexing process allows the introduction of an order relation R in the CB that can be used to compare cases. Indices are organized in the form of a Subsumption Hierarchy.

(CB, R)={ [C_k / k=(1,..,q) and q $\in$ IN], R}={(C_1 , .., C_q)/ idx(C_1) $\subseteq$ .. $\subseteq$ idx(C_q)}

Let us say that two cases C and C'$\in$ CB fulfill the relation

$idx(C) \subseteq idx(C')$ if $\begin{array}{l} idx(S_I) \subseteq idx(S'_I) \\ idx(S_F) \supseteq idx(S'_F) \end{array}$

And it is expressed in terms of their components,

$$idx(S_I) \subseteq idx(S'_I) \rightarrow \begin{Bmatrix} O_r(S_I) \geq O_r(S'_I) & \forall r = 1,\ldots p \\ R_s(S_I) \leq R_s(S'_I) & \forall s = 1,\ldots q \end{Bmatrix}$$

$$idx(S_F) \supseteq idx(S'_F) \rightarrow \begin{Bmatrix} O_r(S_F) \leq O_r(S'_F) & \forall r = 1,\ldots p \\ R_s(S_F) \geq R_s(S'_F) & \forall s = 1,\ldots q \end{Bmatrix}$$

- Definition: Let us say that S(P') is a possible CBR solution of the target P,

  $\forall$ C'= (P' , S(P'), ℑ[S(P')]) / idx(C') $\supseteq$ P

Example: Given three cases, C1, C2, C3, and their indices, where the initial states are null, and just there are values for the final state.

idx(C1)={(O'_1,a_1), (R'_1,b_1), (R'_2, b_2)}= {(O'_1,1.7), (R'_1,95), (R'_2, 21.6)}

idx(C2)={(O'_1,a_1), (R'_1,b_1), (R'_2, b_2)}= {(O'_1,1.1), (R'_1,80), (R'_2, 19.2)}

idx(C3)={(O'_1,a_1), (R'_1,b_1), (R'_2, b_2)}= {(O'_1,0.9), (R'_1,100), (R'_2, 22)}

If the problem to solve may be represented by P=(S_i, S_f) where its solution satisfy,

$$S(P) = \begin{Bmatrix} O'_1 \geq 1 & \\ R'_1 \leq 105 \ , & R'_2 \leq 25 \end{Bmatrix}$$

the relationship idx(C3)$\subseteq$ P$\subseteq$ (idx(C1), idx(C2)) may be established. So the definitions presented above let us know that idx(C3) is not a possible CBR solution of the target P, while idx(C1), idx(C2) are possible CBR solution for the problem P.

Given a canonical coordinate system ℵ =(A_1, A_2,…,A_v) on I(Θ), the set may be reordered, differentiating between:

{F_m}= {$A_{j \text{ with } j\leq v}$ / A_j growing} and {G_n}= {$A_{k \text{ with } k\leq v}$ / A_k decreasing} so,

ℵ= {F_m} $\cup$ {G_n} and m+n=v

Then, giving an i$\in$ I(Θ), a functional dependency relationship may be obtained in terms of the attributes i= i [$e_1(\tau_1, \tau_2,\ldots,\tau_j)$, $e_2(\tau_1, \tau_2,\ldots,\tau_k)$,…,$e_s(\tau_1, \tau_2,\ldots,\tau_q)$)] = = i(τ_1, τ_2,…,τ_n) and in terms of its canonical or state variables:

$i = i(A_1, A_2, \ldots, A_v) = i(F_1, F_2, \ldots, F_m, G_1, G_2, \ldots, G_n)$ which determines a functional relationship of the type $A_j = A_j(\tau_1, \tau_2, \ldots, \tau_n)$.

Example: If we consider now two possible routes i_1 and i_2, together with their values, presented in Table 4.

$i_1 = M1 \wedge D(A,A) \wedge M2 \wedge D(A,A) \wedge R1 \wedge D(A,B) \wedge M3 \wedge D(B,B) \wedge R2$
$i_2 = M2 \wedge D(A,A) \wedge R1 \wedge D(A,B) \wedge M3 \wedge Tx(B,A) \wedge M1 \wedge Tx(A,B) \wedge R2$

Table 3. Values for the intentions i_1 and i_2

Schedule(hr)	**10-11**	**11-12**	**12-13**	**13-14**	**14-16**	**16-18**	**18-20**	**20-21**	**21-22**
intention	M1	D(A,A)	M2	D(A,A)	R1	D(A,B)	M3	D(B,B)	R2
Costs(€)	6	0	6	0	12	0	0	0	12
Time (hr)	1	1	1	1	2	2	2	1	1
Evaluation	1	--	2	--	1	--	2	--	2

Schedule(hr)	**10-11**	**11-12**	**12-13**	**13-15**	**15-17**	**17-19**	**19-20**	**20-21**	**21-22**
intention	M2	D(A,A)	R1	D(A,B)	M3	Tx(B,A)	M1	Tx(A,B)	R2
Costs(€)	6	0	12	0	0	3	12	3	12
Time (hr)	1	1	2	2	2	1	1	1	1
Evaluation	2	--	1	--	2	--	1	--	2

If our coordenates system is represented by $\aleph = (A_1, A_2, A_3, A_4) = (N, T, C, E)$ where N=Places visited, T=Time spent in the visit, C=Cost of the visit, E=Evaluation (visit satisfaction) then the previously presented intentions can be expressed as,

i_1--> N=3, T=12 (h), C=36(€), E=1.6 i_2 --> N =3, T=12 (h), C=48(€), E=2

Now the fundamental relationship between the BDI agents and the CBR systems can be introduced. We define "state ς of an intentional process" and we denote as $\varsigma = \{e_1 \wedge e_2 \wedge \ldots \wedge e_{s-1} \wedge e_s\}$ to describe any of the situations intermediate to the solution $i = \{e_1 \wedge e_2 \wedge \ldots \wedge e_r$, with $r \le s\}$ that admits a representation over $\aleph$. Moreover, the solution S(P) for a given problem $P = (S_I, \{\theta_j\}, S_F)$ can be seen as a sequence of states $S_k = (\{O_r\}_{r=1,\ldots,p}, \{R_s\}_{s=1,\ldots,q})$ interrelated by operators $\{\theta_h\}$.

Given a BDI agent over Θ with a canonical system, $\aleph = (A_1, A_2, \ldots, A_v)$ in the set $I(\Theta)$ that may be reordered as $\aleph = (F_1, F_2, \ldots, F_m, G_1, G_2, \ldots, G_n)$, we establish the relationship between the set of parameters:

$$\{F_m\} \longleftrightarrow \{O_r\} \qquad \{G_n\} \longleftrightarrow \{R_s\}$$

The identification criteria may be established among

- the intentional states, $\varsigma_i \in I(\Theta)$, and the CBR states, $S_k \in T(BC)$.
- and a relationship may be established among the agents desires $I(\Theta)$ and the effectiveness operator $\Im[S(P)]$ of the CBR system.

Then the mathematical formalisation proposed can be used as a common language between agents and CBR system and solves the integration problem. The relationship presented here shows how deliberative agents with a BDI architecture may use the reasoning cycle of a CBR system to generate solutions S(P).

Example: We continue with the previous example, if the values P,T,C,E are represented by a structure

$$S = \left\{ \frac{Or}{Rs} \right\}_{r=1,..,p,..s=1,..,q}$$

then each intention may be considered as a case, with an associated index.

$$idx(C1) = \left\{ \frac{(3,1.6)}{(36,12)} \right\} \qquad idx(C2) = \left\{ \frac{(3,2)}{(48,12)} \right\}$$

If a problem P is presented to the agent in the following terms:

$$S(P) = \left\{ \begin{matrix} N \geq 3 & E \geq 1.5 \\ T \leq 12\,h, & C \leq 50€ \end{matrix} \right\}$$

then idx(C1), idx(C2) are two possible CBR solutions because given the previously presented definition, $P \subseteq$ idx(C1), idx(C2). The desire function

$$d : I(\Theta) \longrightarrow \Omega(\aleph)$$

$$i = (e_1 \wedge \ldots \wedge e_r) \rightarrow F = F(i) = \{ P \geq 3, E \geq 1.5, T \leq 12, C \leq 50 \}$$

may be expressed in terms of $\Im(S(P)) = (\Im_x, \Im_y)$. For this example the values can be claculated using the expressions of section 3.2.

$\Im_x(S(P)) = 0.412$, where S(P) must achieve at least 41.2% of its objectives.

$\Im_y(S(P)) = 0.790$, where S(P) should not require more that 79% of the resources, while the values of the efficiency parameters of cases idx(C1) and idx(C2) are:

$\Im_x(C1) = 0.432$, and $\Im_y(C1) = 0.738$

$\Im_x(C2) = 0.516$, and $\Im_y(C2) = 0.775$

that holds:

$\Im_x(C1) > \Im_x(S(P))$ $\qquad$ $\Im_x(C2) > \Im_x(S(P))$

$\Im_y(C1) < \Im_y(S(P))$ $\qquad$ $\Im_y(C2) < \Im_y(S(P))$

as shown in section 3.3.

The relationship, presented here, shows how deliberative agents with a BDI architecture may use the reasoning cycle of a CBR system to generate solutions S(P). When the agent needs to solve a problem, it uses its beliefs, desires and intentions to obtain a solution. Previous desires, beliefs and intentions are stored taking the form of cases and are retrieved depending on the current desire. Cases are then adapted to generate a proposed solution, which is the agent action plan.

4 Modelling dynamic CBR-BDI agents

The proposed analytical notation allows the definition of "CBR-BDI" agents. Such agents have the ability to plan their actions, to learn and to evolve with the environment, since they use the reasoning process provided by the CBR system. CBR systems may be implemented and automated in different ways [4, 6] depending on the problem which must be solved. This section shows how variational calculus is used in the framework of the CBR system to automate the retrieval stage, which gives the agents more autonomy [8, 10].

4.1 Formalization of the integration of the CBR-BDI agents

The operations that are carried out during the reasoning process of the CBR system are now defined, using the previously introduced notation.

4.1.1 Retrieval and Adaptation

During the retrieval phase, a problem P´ stored in the case base CB and that is similar to the target problem P is identified. Given the problems P and P', it is said that P' is "similar" to P and it is denoted $P' \approx P$, if the case:

- $C' = (P', S(P'), \Im[S(P')]) \in CB$, is a <u>possible CBR solution,</u> and
- $idx(C') \supseteq \{idx(C_k)\ k_{=}(1,...., n)\}$

Now we use the parameter **efficiency** $\varsigma[S(P)]$, that indicates the amount of resources that should be spent to achieve each objective.
The cases for which the efficiency is maximum are selected and denoted by $\varsigma[S(P)]max$, which is a subset of the previously selected solutions:
$P \subseteq idx(C1), idx(C2), idx(C3),\ldots, idx(Cr)$, with $r \leq m$.

Now we need to identify which is the best case from this subset. Before to show how such case may be identified, a non-linearity effect in the relationship between the cases with their attributes is introduced in the following examples.

Example: The visit to a museum M2, with E=1, may cost C=2€, while a museum M1 with E=2, may be visited for free, C=0€ (for example, if there is a public program of cultural promotion).

To incorporate such non-linearity to the problem, all the non linear processes are codified in the function $V=V(A_1,A_2,...A_v) \neq 0$. The function V on $\aleph = (A_1,A_2,...A_v)$ introduces constraints between such variables that can be graphically associated with "curvature" in the phase space, such as the one represented in Figure 2.

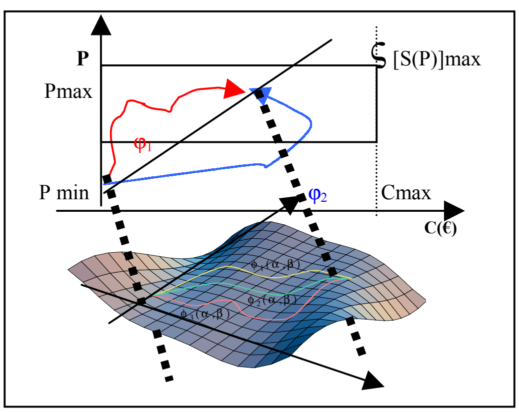

Fig. 2. Effects of non-linearity

In terms of our tourist agent, considering only $\aleph = (A1, A2) = (N, C)$ and given a target problem to solve defined by a P_{min} and a C_{max}, Figure 2 may represent three potential solutions φ1, φ2, φ3, assuming non-linearity effects.

If we represent the cases stored in a space of coordinates $\aleph$ = (A1, A2,…,Av), the stored cases define a hyper-surface (if we extrapolate the lattice of cases to a continuous surface) and each case can be represented by a curve on that surface. The advantage of modelling the cases as a hyper-surface is that we can apply on the cases a variational calculus based strategy.

Table 4. Believes: Instance characterisation.

Class	Monument	
Entity	New Cathedral	
Visiting time	10-13 h	
Time for a visit	1 h	
Visiting Cost	6 €	
Zone	A	
Eval.	Mo1	1,13
	Mo2	2,85
	Mo3	2,76
	Mo4	1,12
RF	0,3	

(a)

Class	Spectacle	
Entity	Local music band	
Visiting time	21-23 h	
Time for a visit	2 h	
Visiting Cost	3 €	
Zone	B	
Eval.	Sp1	2,20
	Sp2	2,34
	Sp3	2,49
	Sp4	1,46
RF	0,5	

(b)

Class	Journey	
Entity	Bus / D(B,A)	
Time	24 h., at 10min intervals	
Time for a visit	15 min	
Visiting Cost	1 €	
Zone	A	
Eval.	Jo1	2,12
	Jo2	1,01
	Jo3	1,89
	Jo4	2,35
RF	0,9	

(c)

Class	Restaurant	
Entity	Tapas Bar	
Lunch time **Dinner time**	13-16 h / 22-23:30 h	
Time for a visit	1 h	
Visiting Cost	12 €	
Zone	A	
Eval.	Re1	2,56
	Re2	2,23
	Re3	1,41
	Re4	2,29
RF	0,9	

(d)

For example, Table (4.a) refers to the New Cathedral of the City of Salamanca, which may be visited from 10:00 to 13:00. The average time for a visit is one hour and the cost is 6 Euro. It is situated in the Zone A (the city of Salamanca has been divided into 5 different areas: A to E). The profiles of the visitor to Salamanca have been divided in: Mo1 (cultural tourist), Mo2 (art expert), Mo3 (family visit) and Mo4 (generic tourist) with respect to the monuments. The classifications may vary with respect to other entities. Each beliefs maintains information related to the evaluation provided by the tourists after the visit. The evaluation (between 0 and 3) is averaged taking into consideration the group to which the tourist belongs. The risk factor, RF, provides information about the probability of finding a similar item to a given one, in case it may not be visited if it is a monument or if it is fully booked in the case of a restaurant for example.

Table 5: Intentions evaluation

Intention	Tourist	C(€)	T(h)	No.Places	Eval: Mo1-Mo4		Eval: Sp1-Sp4
I1	T1	18	12	5	Mo1: 1,12	..	Sp1: 2,25
I1	T2	18	13	5	Mo3: 2,30	..	Sp3: 2,50
I2	T3	21	11	4	Mo2: 2.38	..	Sp2: 2,50
----		----	----		----	--	----
In		..	..		..	..	..

Table 5 shows a table maintained by the agent that associates to each route or intention its total cost, the time required by the tourist to carry it out, the characterisations of the tourist, its evaluation with respect to such characterisations and the average evaluation. For example, the route/intention, I1, was carried out by the tourist T1, the total cost was 18 Euro, the time spent on it was 12 hours, the tourist was doing cultural tourism (Mo1), he enjoys traditional music (Sp1) and he has evaluated his interest in the monuments visited in this route as Mo1=1,12, and of the spectacles attended as Sp=6,25. The agent may then use this information to retrieve past intentions taking into consideration the preferences of the tourist.

Then, for simplification purposes, we may represent the routes in function of the coordinates (A1,A2,A3,.. ,An), where for example:

- A1=Cost (€)=C= it is a monotonically increasing variable (it accumulates the costs taken step by step)
- A2=No. Places =P=number of visited items. It is an accumulative variable
- A3=Time (hr)=T= monotonically increasing variable as above.
- A4=Evaluation =E=mean of the quality. A priori we cannot establish a defined tendency.

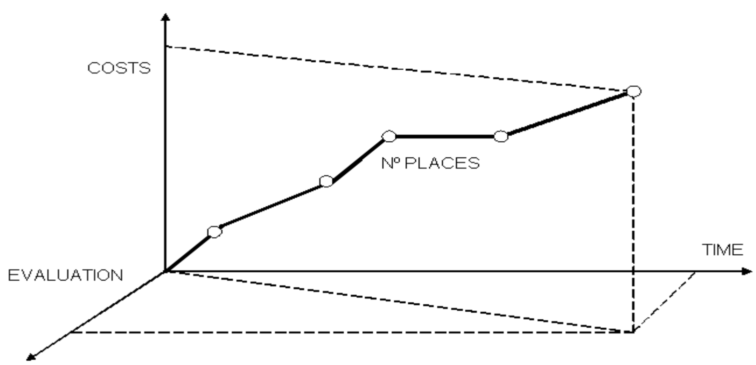

Figure 3: Graphical characterisation of a case.

In dynamic environments with uncertainty it is difficult to guarantee that a given algorithm retrieves the best cases from the case base, and the evaluation, in real time, of all the possible options may have unacceptable computational costs. In our proposal the agent first has to interrogate the tourist and obtain information about his desires: time and money to spend in the visit and preferences with respect to art, food, accommodation, etc. Figure 3 presents a graphical characterisation of a simplified historical case. The agent then applies variational calculus to such retrieved cases to obtain the solution closest to the optimum solution [10]. Let see now this process' works in detail.

Step 1: The solution has to be found in the retrieved cases that satisfy the selection criteria imposed by the tourist. Such a subset defines a topology and therefore to obtain an optimum solution implies defining a metric on the subset. For example, if we supposes that the visit to the monuments, the transport, etc. is free after 12:00, the cost-time relationship may be represented by Figure 4.

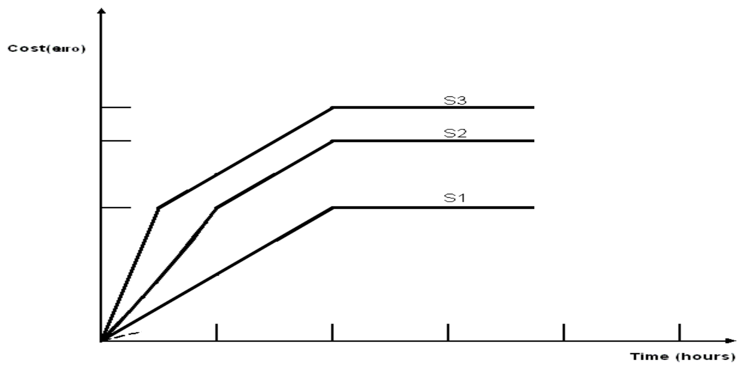

Figure 4: Cost-time (C-T) relationship

The framework of this problem implies that the items are not crossed increasing the cost progressively.. Then the agent solution needs to be based in the retrieved routes, which have been carried out by tourists with similar preferences and profiles that the tourist interested in carrying out a tour.

Step 2: On the hyper-surface generated by the retrieved solutions -on the space defined by the variables (A1, A2,..., An)– variational calculus is applied [8]. Now it will be shown how variational calculus can be used to automate the reuse process. Let us consider a case base (CB, R)={[C_k / k=1,.,q and q $\in$ IR],R} and the set of attributes of the case base T(CB)= (α_1, α_2,,..., α_m,), α_j ϵ T. Using the relationships between BDI agents and CBR systems established, it is denoted T(BC)=(A_1, A_2,...A_v), coordinates system of I(Θ), which allows us to define a function V on the space I(Θ), that stores the information of all the cases Ck ϵ CB.

$$V : T(CB) \longrightarrow T(CB)$$
$$(A_1, A_2, \ldots A_v) \longrightarrow V(A_1, A_2, \ldots A_v)$$

If we consider two states (Si , Sf) initial and final, on I(Θ), the function V shows all the intentions i ϵ I(Θ), that joins both states (S_i ,S_f) and that has related a case C_k ϵ CB**.** On the phase space, the function V=V(A_1,A_2,...A_v) is translated onto a surface Π_0[A_1,A_2,...A_v]=0, where the notion of Euclidean distance is defined.

Let S_i , S_f be two states (two points on I(Θ), then D (S_i , S_f) takes the form

$$D(S_i, S_f) = \sqrt{(A_{i1} - A_{f1})^2 + (A_{i2} - A_{f2})^2 + .. + (A_{im} - A_{fm})^2}$$

where S_i =(A_{i1}, A_{i2}, , A_{iv}), S_f = (A_{f1}, A_{f2} , , A_{fv})

In the m=3 case, and with A_1=X, A_2=Y, A_3 =Z, the theory of variational calculus says that a coordinate system (λ, μ) exists which allows an expression of the functional F=F(λ, μ), that associates to each curve between S_i and S_f on Π_0[x,y,z]=0 with its length, thus we can obtain a solution of

$\frac{\partial F}{\partial \mu} - \frac{d}{d\lambda}\left(\frac{\partial F}{\partial \mu'}\right) = 0$; that we call $\mu = \mu_0(\lambda)$ and that takes the form χ_0 = χ_0[x,y,z] on the original coordinates (X, Y, Z). This function is named the geodesical curve.

Step 3: Solutions of differential equations in variational problems exist only in exceptional cases. In the actual problem, the routes are non-differentiable broken lines. In these cases, the variational problem is just a theoretical boundary for function optimisation problems with a finite number of variables. A differentiable continuous functional V[y(x)] can be expressed as a function of a Taylor series or a Fourier series, taking the following form:
$V[y(x)] = V[a_0+a_1x_1+a_2x_2+...]$ So we can deal with the problem in a similar way and study which is the optimum of a function depending of a finite number of variables. This mechanism is known as method of finite differences, and develops an equivalent equation to Euler's system for broken functions, as represented in Figure 5.

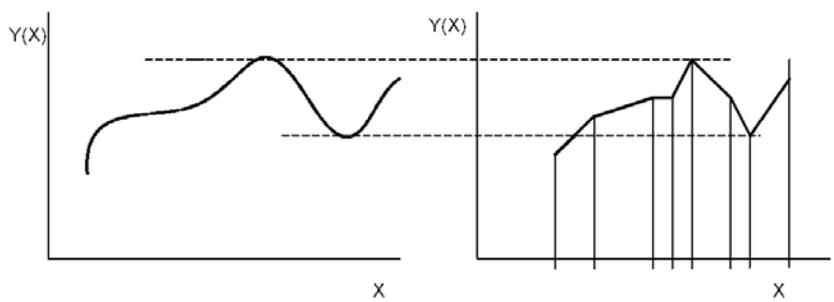

Figure 5: Adaptation of a continuous function by a discrete one

Therefore, variational calculus with mobile frontiers is used [8]. Variational calculus with mobiles frontiers calculates the optimum solution taking into consideration that one extreme is moving over a function:
f=f(A1, A2,...,An), as represented in Figure 6.

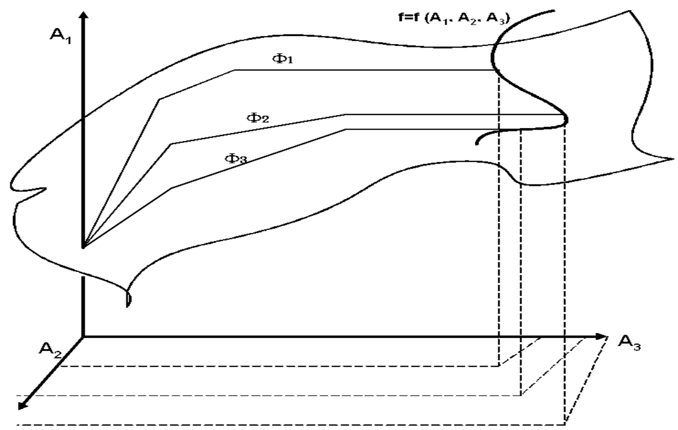

Figure 6: Graphical representation of three intentions/routes in a three dimensional space.

A generalisation of Euler´s equation exists valid for any number of parameters. In this case, the solution is obtained solving a n-dimensional Euler´s system of differential equations. In the most general case, the mapping $V=V(A_1, A_2,..., A_m)$ generates curves that cannot be differentiated because V only takes values at discrete points corresponding to defined and stored cases.

Let us now define a mapping σ, as $\boldsymbol{\sigma = \chi_0 - \psi}$**,** where $\boldsymbol{\chi_0}$ is the solution obtained by Euler´s equation [8] and $\boldsymbol{\psi} \in \{\varphi(S_i,S_f)\}$ is a path between S_i, and S_f, stored in the case-base as a case C ϵ CB. Then we will call "the closest to the optimal curve $\boldsymbol{\psi_0}$" the mapping of $\{\varphi(S_i,S_f)\}$ given by the minimisation of

$$I = \int_{ei}^{ef} \{\sigma [X,Y,Z]\} \, dx \, dy \, dz$$

where $\boldsymbol{\psi_0 = \{ S_i = S_0^{(0)} , S_1^{(0)}, S_2^{(0)}, S_3^{(0)}, S_4^{(0)},...., S_m^{(0)}, ..., S_s^{(0)} = S_f \}}$, and S_k are the states obtained to achieve the solution.

So far it has been shown how variational calculus can be used to select the closest to the optimum curve. Variational calculus may then be used to select and retrieve the most appropriate case during the retrieval stage. The retrieved case is characterised as being the one that, in each of its stages, maintains the efficiency most constant.

During the adaptation phase, the system executes a transformational reasoning mechanism [1], that can be represented by the adaptation function A,

$$A : (CB) \times \Sigma(P) \rightarrow C$$

$$(C \quad , \quad P) \quad \rightarrow A[S(P'), P] = \{ P, A[S(P')], \Im(A[S(P')])\}$$

with $P \in \Sigma(P)$ is called set of problems, and $C=(P',S(P'), \Im([S(P')])$

In [2] a retrieval mechanism that identifies a case easy to adapt is suggested. Therefore the retrieval mechanism should be subordinate to the adaptation one. In our proposal we assign higher relevance to the retrieval strategy. If P=(Si, Sf) and during the retrieval stage it is obtained $C'=\{P',S(P'), \Im[S(P')]\}) \in (CB)$, the adaptation function constructs a solution for P maintaining the sequence of operators that S(P'). If at any point the sequence may not be applied, a new retrieval cycle is initiated from the state in which the sequence was interrupted. Therefore the adaptation function can be seen as a serie of operators:

$A= \alpha_m \bullet \alpha_{m-1} \bullet ...... \bullet \alpha_2 \bullet \alpha_1$, where each operator is a part of a retrieved case.

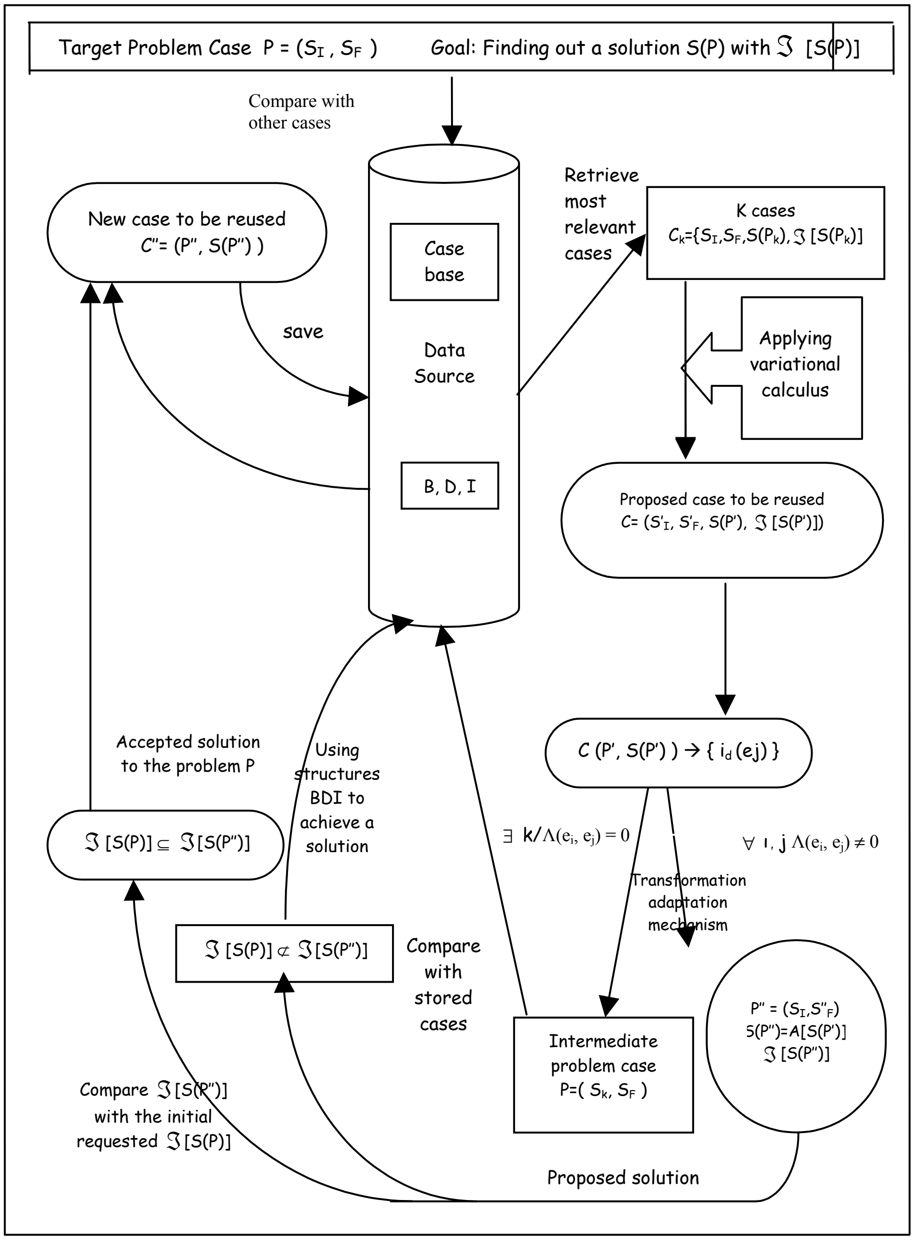

Fig. 7. Formal Model Detailed Schema

Figure 7 shows how variational calculus is applied during the retrieval stage to select the closest to the optimum case from the case-base. In this figure we show graphically the working and information flow during the reasoning process of the agent, introduced in this section.

Example: Given a problem $P=(S_I, S_F)$, and the retrieved case $P'=(S'_I, S'_F)$ with $S(P')=(S_1=S'_I, \theta'_1, S'_2, \theta'_2, ..., \theta'_n, S'_{n+1}=S'_F)$.
The first state of the adapted solution $A[S(P')] = \{ S_I=S_1, \theta'_1, S''_2, ..., \theta'_i, S''_{i+1}\}$ for the problem P is the initial state of P, and uses the operators θ'_i of S(P') until the state S''_{i+1}, from which it can not progress. This is an incomplete solution for the problem P, which is denoted by $P1=(Si, S''_{i+1})$. So $\alpha 1[S(P')]= \{S_I, \theta'_1, ..., \theta'_j, S''_{j+1}\} = S(P1)$.

This process is repeated again for the problem $P''=[S''_{i+1}, S_F]$, whose initial stage is S''_{i+1}. So $P''=[S''_{i+1}, S_F]$, and then a new case has to be found that allows the plan to progress and to reach the state S where $S''_{i+1} \leq S \leq S_F$. If we denote $P2=[S''_{j+1}, S]$, the operator $\alpha 2[S(P'')] = S(P2)$ may be identified.

This process is again repeated until a final state S_F is found, and then,
$A[S(P')]= (\alpha_m \bullet ... \bullet \alpha_2 \bullet \alpha_1)[S(P')]= (\alpha_m \bullet ..... \bullet \alpha_2)\ [S(P'')]= ...=(\alpha_m)\ [S(P^m)]= S(P)$

4.1.2 Revision and Memorisation

In this phase the case solution generated in the previous phase is evaluated and reviewed. A problem P occurs for which we want to obtain a solution S(P) with $\Im[S(P)]$. If, during the retrieval step, a case $C=(P', S(P'), \Im[S(P')])$ is recovered and the adaptation step ensures a solution $S(P)=A[S(P')]$, the review must guarantee that:

$$\Im\{ A[S(P')]\} \supseteq \Im [S(P)]$$

The problem target and the characteristics of the adapted solution can be memorized as a new case to be reused in the future and is denoted by
$C= \{ P, A[S(P')], \Im(A[S(P')]) \} = (P, S(P), \Im[S(P)])$

4.2 Planning with variational calculus

This section shows how the variational calculus, introduced in the previous section, allows the agents to plan and replan at execution-time because this formalism is used to select the most adequate case during the reuse phase of the reasoning process to solve a given problem. Assuming that potentially significant changes can be determined after executing a primitive action, it is possible to control the dynamism of the new events of the domain and thus achieve an appropriate reconsideration of the problem [8].
Variational calculus may also deal with dynamic problems such as this one. When the plan proposed by the agent is stopped for any reason (i.e. the tourist may decided to spend more time visiting a monument, have a longer lunch, etc.), variational calculus calculates a new plan. In this case the new initial state is the point at which the initial proposed route has stopped, as shown in Figure 8.

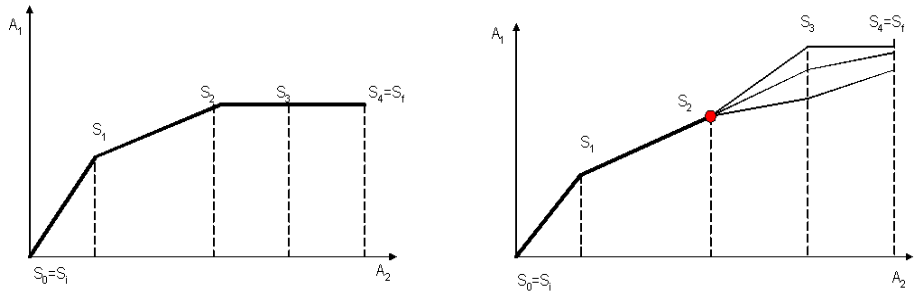

Figure 8: Replanning at execution time.

If it is accepted that the environment changes, it is also necessary to define a reasoning mechanism capable of dealing with such changes by modifying the initial desires and intentions.

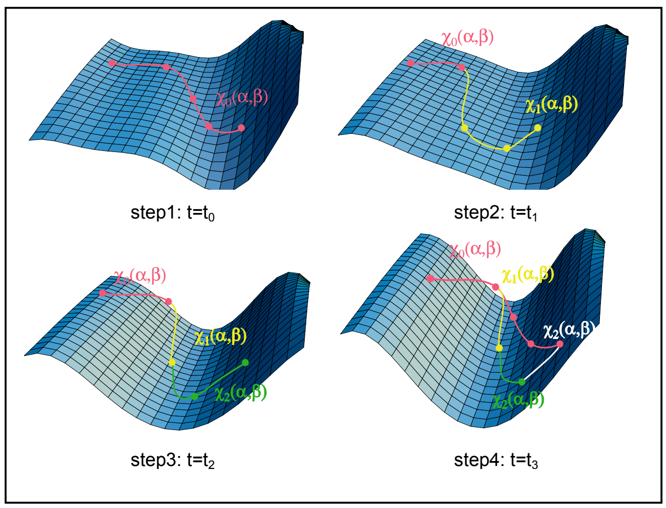

Fig. 9. 3D representation of a dynamic environment.

Nevertheless the reasoning process may be maintained since the general description problem remain constant. If at $t_{0,}$ the function V(X, Y, Z) takes the form denoted by $V_0(X,Y,Z)$, at t_1, V is denoted by V_1 (X, Y, Z), with the associated surface Π_1 (X, Y, Z) = 0 on the phase space, upon which it is possible to obtain the optimal curve between two new points, S_i and S_f where $S_i = S_1^{(0)}$, and $S_1^{(0)}$ is the second state of $\psi_0 = \{ S_i = S_0^{(0)}, S_1^{(0)}, \ldots, S_s^{(0)} = S_f \}$ and S_f is the final state or solution state of the global problem.

Solving the Euler´s equations, $\boldsymbol{\chi_1 = \chi_1(X, Y, Z)}$ is obtained, which may be used to calculate an expression for $\boldsymbol{\psi_1}$, denoted as $\boldsymbol{\psi_1 = \{ S_i = S_1^{(0)}, S_1^{(1)}, S_2^{(1)}, S_3^{(1)}, S_4^{(1)}, \ldots, S_m^{(1)}, \ldots, S_s^{(1)} = S_f \}}$ and the same can be done for any $\mathbf{t_j}$ (see Figure 9).

From the previous equations, and based on variational calculus tools, an expression can be determined to identify the final solution of the CBR-BDI agent. This expression, which represents the agent plan, can be obtained in execution-time and takes the following form:

$$\Psi_{final} = \begin{cases} \Psi_0,\ldots K \quad t \in (t_0, t_1) \\ K\ K\ K\ K\ K\ K\ K\ K \\ \Psi_{s-1}, K \quad t \in (t_{s-2}, t_{s-1}) \\ \Psi s,.. K \quad t \in (t_0, t_1) \end{cases}$$

6 Tourist guide usal: A "CBR-BDI" system to solve problems in the e-tourism domain

The tourism industry is an information intensive economic sector. This activity, as many others, requires the use of a great amount of data, ranging from product data to technical publications, from tourism regulations to best practice guides. A multiagent based system has been developed for guiding tourists around the city of Salamanca. The agent based system can be accessed via Internet or wireless devices such as mobile phones, PDAs, etc. The system is composed of a CBR-BDI agent that advises tourists and that communicate with other agents that maintain uptodate information about Salamanca, its monuments, restaurants, spectacles, etc. This paper shows how the CBR-BDI agent identifies adequate routes for tourists based on previous experiences. The agent is therefore capable of determining plans using stored cases or experiences and of learning from them. Taking into consideration the characteristics of the present problem, the CBR system embedded in the agent needs to:

- generate plans or tourist routes
- handle large amounts of contextual information, in real time and using temporal reasoning
- re-plan in execution time
- incorporate new knowledge (in the form of new beliefs, or new experiences making cases) and to learn from its experiences after successes or failures in its advice.

Figure 10 describes the interaction process between the user and the tourist guide agent. The tourist may use a mobile device to contact the agent, and then introduces his/her login and password, and indicates to the agent his/her preferences (monuments to visit, visits duration, time for dinner, amount of money to spend, etc.).The agent then generates a plan for the user according to his/her preferences and sends it back to him/her.

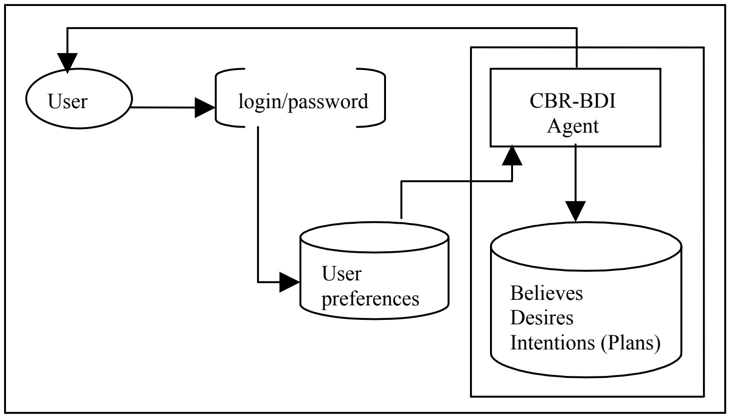

Fig. 10. Schema of relations in TOURISTGUIDE-USAL

When the agent is contacted by the tourist, it receives information about his desires and preferences. In this section we are going to see how the agent reasons and provides the solution to the tourist using a particular case.

The tourist desires to spend a day visiting Salamanca (12 hours visit), he is an art expert, and wants to visit the Museum of Contemporary Art (about which he has heard of) and eat in fast food restaurants. He does not want to spend more that 60 Euro, and he wants to visits monuments, restaurants, etc. that have been evaluated positively with a value upper of 1,3 (evaluation range 0-3).

Table 6: Retrieved cases.
(Where M1...M8 correspond to Monuments, D(X,Z) correspond to distances and R1…R6 correspond to restaurants).

$i_1 = M1 \wedge D(A,A) \wedge M2 \wedge D(A,A) \wedge R1 \wedge D(A,B) \wedge M3 \wedge D(B,B) \wedge R2$

Schedule(hr)	**10-11**	**11-12**	**12-13**	**13-14**	**14-16**	**16-18**	**18-20**	**20-21**	**21-22**
Intention 1	M1	D(A,A)	M2	D(A,A)	R1	D(A,B)	M3	D(B,B)	R2
Costs(€)	6	0	6	0	12	0	0	0	12
Time (hr)	1	1	1	1	2	2	2	1	1
Evaluation	1	--	2	--	1	--	2	--	2

attributes	
C(€)	36
T(hr)	12
E	1.6

$i_2 = M2 \wedge D(A,B) \wedge M3 \wedge D(B,B) \wedge M7 \wedge D(B,A) \wedge R3 \wedge D(A,A) \wedge R1$

Schedule(hr)	**10-11**	**11-12**	**12-14**	**14-15**	**15-16**	**16-18**	**18-20**	**20-21**	**21-22**
Intention 2	M2	D(A,B)	M3	D(B,B)	M7	D(B,A)	R3	D(A,A)	R1
Costs(€)	6	0	6	0	0	0	12	0	12
Time (hr)	1	1	2	1	1	2	2	1	1
Evaluation	2	--	2	--	1	--	2	--	1

attributes	
C(€)	36
T(hr)	12
E	1.6

$i_3 = M8 \wedge D(A,A) \wedge M1 \wedge D(A,A) \wedge R5 \wedge D(A,A) \wedge M4 \wedge D(A,B) \wedge R2$

Schedule(hr)	**10-11**	**11-12**	**12-13**	**13-14**	**14-16**	**16-17**	**17-19**	**19-21**	**21-22**
Intention 3	M8	D(A,A)	M1	D(A,A)	R5	D(A,A)	M4	D(A,B)	R2
Costs(€)	3	0	6	0	18	0	6	0	12
Time (hr)	1	1	1	1	2	1	2	2	1
Evaluation	1	--	1	--	2	--	8	--	2

attributes	
C(€)	45
T(hr)	12
E	1.6

i_4=M1 ∧ D(A,A) ∧ M2 ∧ D(A,B) ∧ R4 ∧ D(B,B) ∧ M7 ∧ D(B,B) ∧ R2

Schedule(hr)	**10-11**	**11-12**	**12-13**	**13-15**	**15-17**	**17-18**	**18-20**	**20-21**	**21-22**
Intention 4	M1	D(A,A)	M2	D(A,B)	R4	D(B,B)	M7	D(B,B)	R2
Costs(€)	6	0	6	0	16	0	3	0	12
Time (hr)	1	1	1	2	2	1	2	1	1
Evaluation	1	--	2	--	1	--	1	--	2

attributes	
C(€)	43
T(hr)	12
E	1.4

i_5=M3 ∧ D(B,B) ∧ R2 ∧ D(B,B) ∧ M7 ∧ D(B,A) ∧ M1 ∧ D(A,A) ∧ R3

Schedule(hr)	**10-12**	**12-13**	**13-14**	**14-15**	**15-16**	**16-18**	**18-19**	**19-20**	**20-22**
Intention 5	M3	D(B,B)	R2	D(B,B)	M7	D(B,A)	M1	D(A,A)	R3
Costs(€)	6	0	12	0	0	0	6	0	12
Time (hr)	2	1	1	1	1	2	1	1	2
Evaluation	2	--	2	--	1	--	1	--	2

attributes	
C(€)	36
T(hr)	12
E	1.6

i_6=M2 ∧ D(A,B) ∧ M7 ∧ D(B,B) ∧ R2 ∧ D(B,B) ∧ M3 ∧ D(B,B) ∧ R6

Schedule(hr)	**10-11**	**11-13**	**13-14**	**14-15**	**15-17**	**17-18**	**18-20**	**20-21**	**21-22**
Intention 6	M2	D(A,B)	M7	D(B,B)	R2	D(B,B)	M3	D(B,B)	R6
Costs(€)	6	0	0	0	12	0	6	0	18
Time (hr)	1	2	1	1	2	1	2	1	1
Evaluation	2	--	1	--	2	--	2	--	2

attributes	
C(€)	42
T(hr)	12
E	1.8

The agent retrieves from the case base the cases that satisfy these requirements. If for example, the cases retrieved by the agent are the ones showed in Table 6 and graphically represented in Figure 11, the retrieved instances define the space shown in Figure 12, to which variational calculus with mobile frontiers may be applied (reuse stage) to calculate the optimum solution.

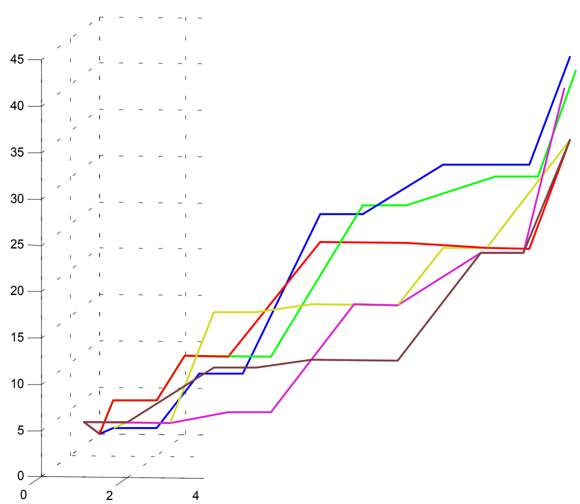

Figure 11: Retrieved instances.

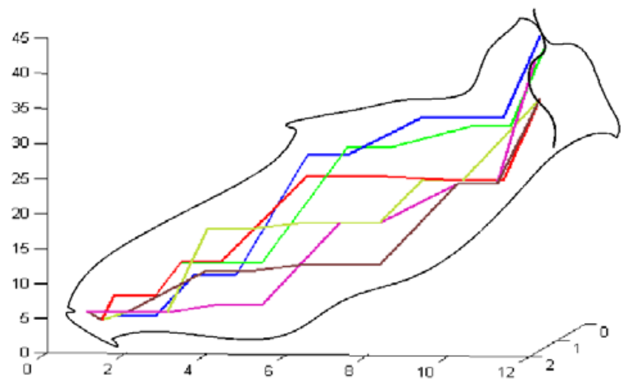

Figure 12: Surface or space to which variational calculus with mobile frontiers may be applied.

Given the optimum solution, the agent calculates which of the retrieved routes is the nearest to the optimum. This will be the proposed route. Figure 13 shows the optimum solution and the selected one.
In this case ,
i_6=M2 ∧ D(A,B) ∧ M7 ∧ D(B,B) ∧ R2 ∧ D(B,B) ∧ M3 ∧ D(B,B) ∧ R6.

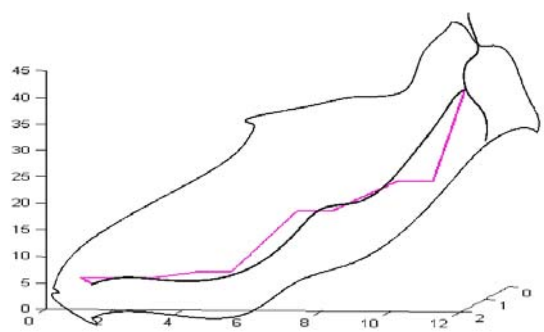

Figure 13: Optimum and closest to the optimum route.

Let see what may happen when the tourist demands a change in the route after having lunch (R2), for any reason. The agent needs to take into consideration the initial constraints together with new ones: there is a new initial state and previously visited monuments should not be visited again. New intentions are retrieved and variational calculus is again applied. Table 7 presents the part of the plan that has already been carried out and Table 8 shows the retrieved intentions that will be use for replaning and generating an alternative solution.

Table 7: Part of the plan already carried out.

i_6=M2 ∧ D(A,B) ∧ M7 ∧ D(B,B) ∧ R2 ∧ ….

Schedule(hr)	**10-11**	**11-13**	**13-14**	**14-15**	**15-17**
intention	M2	D(A,B)	M7	D(B,B)	R2
Costs(€)	6	0	0	0	12
Time (hr)	1	2	1	1	2
Evaluation	2	--	1	--	1

attributes	Partial values
C(€)	18
T(hr)	7
E	1.6

Table 8: Retrieved intentions together with their values.

D(B,A) ∧ M1 ∧ D(A,A) ∧ R7

Schedule(hr)	17-19	19-20	20-21	21-22
intention	D(B,A)	M1	D(A,A)	R7
Costs(€)	0	6	0	20
Time (hr)	2	1	1	1
Evaluation	--	1	--	2

attributes	Partial values	attributes	Total values
C(€)	26	C(€)	44
T(hr)	5	T(hr)	12
E	1.5	E	1.6

D(B,B) ∧ M5 ∧ D(B,B) ∧ R4

Schedule(hr)	17-18	18-19	19-20	20-22
intention	D(B,B)	M5	D(B,B)	R4
Costs(€)	0	6	0	16
Time (hr)	1	1	1	2
Evaluation	--	1	--	2

attributes	Partial values	attributes	Total values
C(€)	22	C(€)	40
T(hr)	5	T(hr)	12
E	1.5	E	1.6

D(B,A) ∧ M8 ∧ D(A,A) ∧ R1

Schedule(hr)	17-19	19-20	20-21	21-22
intention	D(B,B)	M8	D(A,A)	R1
Costs(€)	0	3	0	12
Time (hr)	2	1	1	1
Evaluation	--	1	--	1

attributes	Partial values	attributes	Total values
C(€)	15	C(€)	33
T(hr)	5	T(hr)	12
E	1	E	1.6

Again, the new routes may be represented, see Figure 14, and variational calculus may be applied to obtain the optimum route. The route closest to the optimum is then selected, in this case: D(B,A) ∧ M8 ∧ D(A,A) ∧ R1. Joining both parts of the route can be obtained:
i=M2 ∧ D(A,B) ∧ M7 ∧ D(B,B) ∧ R2 ∧ **D(B,A)** ∧ **M8** ∧ **D(A,A)** ∧ **R1**, as shown in Figure 14.

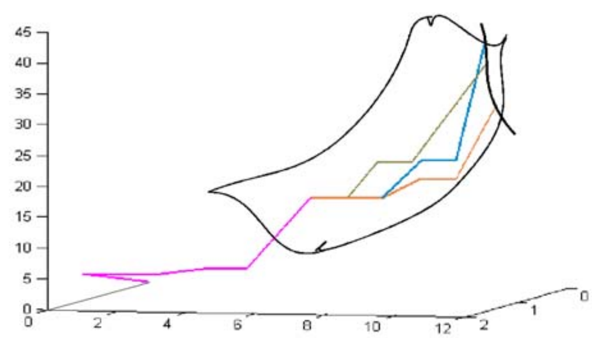

Figure 14. Routes retrieved for replanning.

The tourist evaluates the route after the visit to Salamanca and this information is stored by the agent in its case base.

7. Results and conclusions

The system, here presented, has been tested from the 1st of June to the 15th of September 2002. The case base was initially filled with information collected from the 1st of February to the 25th of May 2002. Local tourist guides provided the agent with a number of standard routes and distributed among his clients Mobile phones, from which they could contact the agent and inform it about the progress of their plans: routes, times, evaluations, etc. During this period the agent stored in its memory 540 instances. Which covered a wide range of all the possible options that offers the City of Salamanca. The system was tested during 115 days and the results obtained were very encourages. Three hotels of the City offered the option to their 4216 guests to use the help of the agent or a professional tourist guide, 7% of them decided to use to agent based system and 28% of them used the help of a tourist guide. The rest of the tourists visited the city by themselves. In this initial experiment the agent intentions were related to a one-day route (a maximum of 12). Therefore the agent provided plans for one day.

On the arrival to the hotel the tourist were asked to evaluate their visit and the route. Table 9 shows the responses given by the tourist after their visit. The tourist that used the help of the software agent provided the answer directly to the agent.

Table 9: Tourists evaluation.

Tourists that	**Number of tourists**	**%**	**Evaluation - degree of satisfaction**				
	4216 (total)		8-10	6-8	4-6	0-4	No answer
Used the help of the agent	295	7%	165 (55,9%)	14 (4,7%)	7 (2,4%)	2 (0,7%)	107 (36,3%)
Used the help of a tourist guide	1180	28%	740 (62,7%)	231 (19,6%)	105 (8,9%)	12 (1%)	92 (7,8%)
Did not use any of the previous	2741	65%	458 (16,7%)	230 (8,3%)	32 (1,2%)	5 (0,2%)	2160 (78,8%)

Table 9 shows the degree of satisfaction of the tourists. As it can be seen, the degree of satisfaction of the tourist that used the help of a professional tourist guide is higher that in the other two cases. Nevertheless the percentage of the tourist which degree of satisfaction was very high (between 8 and 10) is very similar in the case of the tourist that use the help of the agent and in the case of the tourist that use the tourist guide. 38% of the tourist that used the agent based system let us know that the system did not work successfully due to technical reasons (possibly the server was down, there was a luck of coverage, the tourist did not use the wireless system adequately, etc.) If we take this into consideration, we can say that most of the tourist (92%) that used the help of the agent and did not have technical problems had a high or very high degree of satisfaction (6-10). This degree of satisfaction is higher that the one of the tourist (82,3%) that used the help of a tourist guides.

The CBR-BDI architecture solves one of the problems of the BDI (deliberative) architectures, which is the lack of learning capability. The reasoning cycle of the CBR systems helps the agents to solve problems, facilitate its adaptation to changes in the environment and to identify new possible solutions. New cases are continuously introduced and older ones are eliminated. The CBR component of the architecture provides a straight and efficient way for the manipulation of the agents knowledge and past experiences. The proposal presented in this paper reduces the gap that exists between the formalization and the implementation of BDI agents. What we propose in this article is to define the beliefs, desires and intentions clearly (they don't need to be symbolic or completely logic), and to use them in the life cycle of the CBR system, to obtain a direct implementation of a BDI agent.

A mathematical formalism has been introduced to facilitate the representation of BDI deliberative agents and of CBR systems. This analytical formalism also allows the integration of both models and provides a robust framework for the definition and the automatization of the reasoning cycle of the agents, here presented.

Agents need to respond in real time to the user requests and to adapt their solutions in real time, since they inhabit dynamic environments. Variational calculus has been introduced in this paper to facilitate the agents to define their plans and to replan as execution-time in order to provide the best possible service. Variational calculus can be used to obtain the most adequate plan to achieve a goal in environment with uncertainty.

This paper has also shown how the proposed architecture may be used to design an agent for an e-tourism problem. The work presented in this paper is just the first step toward the development of an ambitious framework for developing communities of agents capable of solving problems in an autonomous and intelligent manner. Although the architecture and formalisation described have been applied to the e-tourism domain, we believe it could be also used in any other domain in which agents with learning and adaptation capabilities are required.

Acknowledgements

This work has been partially supported by the CICYT projects TEL99-0335-C04-03 and SEC2000-0249, the PGIDT00 project MAR30104PR and the project SA039/02 of the JCyL.

Bibliography

[1] Aamodt A. and Plaza E. (1994) Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7. No 1, March.

[2] Arnold. V.I. (1971)Ordinary Differential Equations. Springer-Verlag

[3] Bratman M.E. (1987) Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.

[4] Camacho D., Borrajo D. And Molina J. M. (2001) Intelligence Travell Planning: a multiagent planing system to solve web problems in the e-turism domain. International Journal on Autonomous agens and Multiagent systems. 4(4) pp 385-390. December.

[5] Cohen P.R. and Levesque H.J. (1990) Intention is choice with commitment. Artificial Intelligence, 42(3).

[6] Corchado J. M. and Lees B. (2001) A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp105-127.

[7] De Groot M. H. (1970) Optical Statiscal Decisions. McGraw-Hill. New York

[8] Fox, C. An Introduction to the Calculus of Variations. New York: Dover, 1988.

[9] Fyfe C. and Corchado J. M. (2001) Automating the construction of CBR Systems using Kernel Methods. International Journal of Intelligent Systems. Vol 16, No. 4, april 2001. ISSN 0884-8173.

[10] Glez-Bedia, M. and Corchado J. M. (2002) Constructing autonomous distributed systems using CBR-BDI agents. Innovation in Knowledge Engineering, Physica –Verlag (Eds. Faucher, C., Jain, L. And Ichalkaramje, N.). In Press

[11] Jennings N.R. (1992) On Being Responsible. In Y. Demazeau and E. Werner, editors, Decentralized A.I. 3. North Holland, Amsterdam, The Netherlands.

[12] Kinny D. and Georgeff M. (1991) Commitment and effectiveness of situated agents. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91), pages 82-88, Sydney, Australia.

[13] Knobolock C. A., Minton S., Ambite J. L., Muslea M., Oh J. and Frank M. (2001) Mixed- initiative, multisource information asistants. 10th International world wide web conference (WWW10). ACM Press. May 1-5.

[14] Laza R and Corchado J. M. (2001) Creation of Deliberative Agents Using a CBR Model. Computing and Information Systems Journal, Volume 8 No 1, February 2001.ISSN 1352-9404.

[15] Martín F. J., Plaza E., Arcos J. L. (1999). Knowledge and experience reuse through communications among competent (peer) agents. International Journal of Software Engineering and Knowledge Engineering, Vol. 9, No. 3, 319-341.

[16] Olivia C., Chang C. F., Enguix C.F. and Ghose A.K. (1999) Case-Based BDI Agents: An Effective Approach for Intelligent Search on the World Wide Web", AAAI Spring Symposium on Intelligent Agents, 22-24 March 1999, Stanford University, USA

[17] Rao A. S. and Georgeff M. P. (1995) BDI Agents: From Theory to Practice. First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, USA, June.

[18] Shoham Y. (1993) Agent-Oriented programming. Artificial Intelligence, 60(1): pages 51-92.

[19] Watson I. and Marir F. (1994) Case-Based Reasoning: A Review. Cambridge University Press, 1994. The knowledge Engineering Review. Vol. 9. Nº3.

[20] Wendler J. and Lenz M. (1998) CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. Proc. AAAI-98 Workshop on CBR Integrations. Madison (USA) 1998.

[21] Wooldridge M. and Jennings N. R. (1994) Agent Theories, Architectures, and Languages: A Survey. Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages.