Variational calculus based planning for deliberative agentes[(*)]

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

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.

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 [5]. 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.

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. 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 notation used in the referenced works [1] do not have the required degree of expressivity and complexity to introduce differential calculus tools.

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

2.1 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(Θ) is the set of attributes {τ1, τ2,…,τn } 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(Θ) denoted by **e = (τ1, τ2,…,τm)** with m≤ n
We call set of beliefs on Θ and denote ζ(Θ) to the set:
ζ(Θ) = { e=(τ1, τ2,…,τj) / where j =(1,2,…, m ≤ n)}

We introduce the operator "Λ of accessibility" between m beliefs (e_1,e_2,e_3,…,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,…, e_m). If any of the belief (e_1, e_2, e_3,…, 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 ∧ D(A,A) ∧ M3) is possible, and Λ (M1,D(A,A),M3)≠ 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,…,e_s) with s∈ IN and Λ(e_i, e_j)≠ 0
Then, we call set of **intentions** on Θ and denote I(Θ)
I(Θ)= { (e_1, e_2, …, e_k) where k∈ IN }

Now a set of parameters will be associated to the space I(Θ) 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(Θ) any set of linearly independent parameters א =(A_1, A_2,..., A_v) that characterise the elements $i \in$ I(Θ).

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 א =(A_1,A_2)=(N, C). In this coordenates system the following intention:

i_1= M1 ∧ D(A,A) ∧ M2 ∧ D(A,A) ∧ R1 ∧ D(A,B) ∧ M3 ∧ D(B,B) ∧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(Θ)={d: I(Θ)→ $\Omega(\aleph)$ / with I(Θ) set of intentions and $\Omega(\aleph)$ set of mappings on $\aleph$ }

Now, after presenting our definition of the agent's beliefs, desires and intentions, section 2.2 defines the proposed analytical formalism for the CBR system.

2.2 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,{ θ_j }, S_f) with S_i=initial state, and S_f=final state.
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 $\{O_r\}_{r=1,\ldots,p}$ are introduced to express the objectives achieved. The coordinates type $\{R_s\}_{s=1,\ldots,q}$ are introduced to express the resources used. Through these definitions, the parameter effectiveness, ℑ, between two states S and S' can be defined, as a vector ℑ (S, S') = ($ℑ_x$, $ℑ_y$) which takes the form

$$ℑ_x = \frac{O_r(S') - O_r(S)}{O_r \max} \qquad ℑ_y = \frac{R_s(S) - R_s(S')}{R_s \max}$$

The definition implies that ($0 \le ℑ_x \le 1$) and ($0 \le ℑ_y \le 1$).

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 this domain, a case C is a 3-tuple *{P, S(P), $\Im$[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.

2.3 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. 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.

Then, it is said that S(P') is a possible CBR solution of the target P, if

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

Given a canonical coordinate system $\aleph=(A_1, A_2,\ldots,A_v)$ on $I(\Theta)$, the set may be reordered, differentiating between:
$\{F_m\}= \{A_{j \text{ with } j\leq v} / A_j \text{ growing}\}$ and $\{G_n\}= \{A_{k \text{ with } k\leq v} / A_k \text{ decreasing}\}$ so,
$\aleph = \{F_m\} \cup \{G_n\}$ and m+n=v

Then, giving an $i \in I(\Theta)$, a functional dependency relationship may be obtained in terms of the attributes $i= i [e_1(\tau_1, \tau_2,\ldots,\tau_j), e_2(\tau_1, \tau_2,\ldots,\tau_k),\ldots,e_s(\tau_1, \tau_2,\ldots,\tau_q)] =$
$= i(\tau_1, \tau_2,\ldots,\tau_n)$ 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 $A_j = A_j(\tau_1, \tau_2,\ldots,\tau_n)$.

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 $r \leq s\}$ 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, $\aleph=(A_1, A_2,..., A_v)$ 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:

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

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). 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.

3 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 [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 [3, 4].

3.1 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.

3.1.1 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 possible CBR solution, and
- $idx(C') \supseteq \{idx(C_k)\ k=(1,....,n)\}$

Now we need to identify which is the best case from this subset. If we represent the cases stored in a space of coordinates $\aleph = (A1, A2,\ldots,Av)$, 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.

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.

Table 3. 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

(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

(b)

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.

Table 4: 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
----		----	----		----	--	----

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.

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. 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 1.

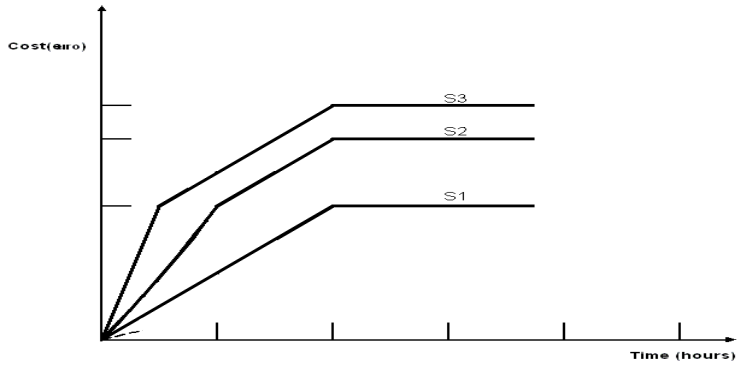

Figure 1: Cost-time (C-T) relationship

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 [3, 4]. 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}. 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.

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

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 $\Pi_0[A_1,A_2,...A_v]=0$, where the notion of Euclidean distance is defined. 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 $\Pi_0[x,y,z]=0$ 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 μ = μ_0 (λ) and that takes the form

$\chi_0 = \chi_0[x,y,z]$ 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, taking the following form: $V[y(x)] = V[a_0+a_1x_1+a_2x_2+...]$.
Therefore, variational calculus with mobile frontiers is used [3].

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 2.

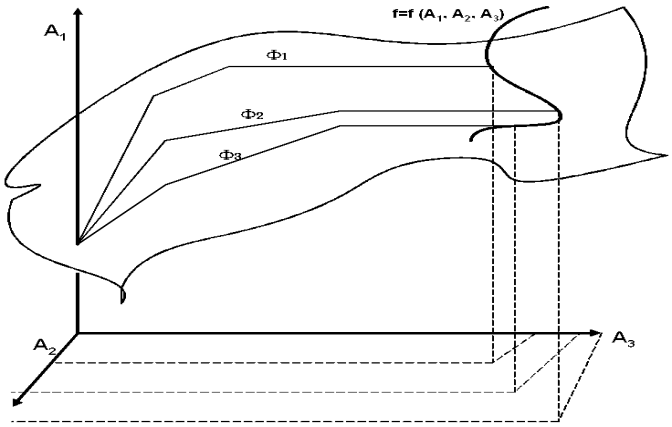

Figure 2: Graphical representation of three intentions/routes in a three dimensional space.

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 $\sigma = \chi_0 - \psi$, where χ_0 is the solution obtained by Euler´s equation [4] and $\psi \in \{\varphi(S_i,S_f)\}$ 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 ψ_0" the mapping of $\{\varphi(S_i,S_f)\}$ given by the minimisation of

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

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. [referrencia]

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

$$A : (CB) \times \Sigma(P) \rightarrow C$$
$$(C \ , \ P) \rightarrow A\,[S(P'), P] = \{ P, A[S(P')], \Im(A[S(P')])\}$$

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.

3.1.2 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)])

3.2 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. 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 3.

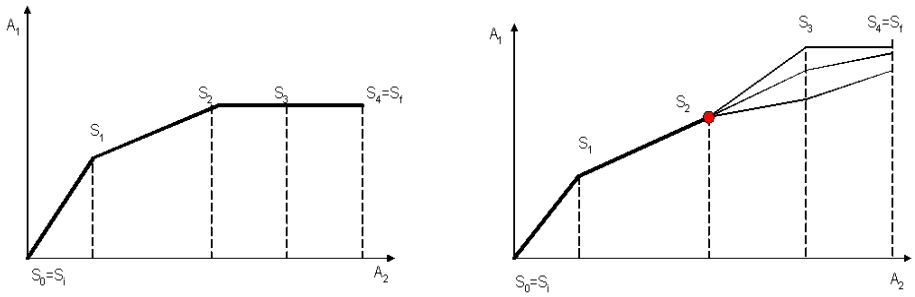

Figure 3: Replanning at execution time.

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:

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

4. A "CBR-BDI" planner 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. 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). The agent retrieves from the case base the cases that satisfy these requirements that they are graphically represented in Figure 4.

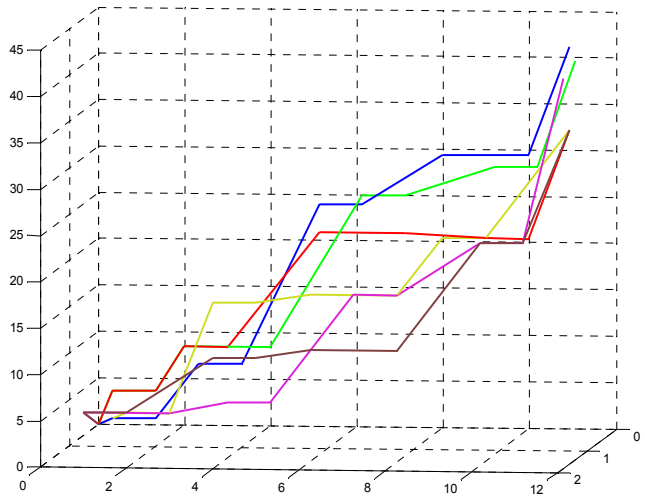

Figure 4: Retrieved instances.

The retrieved instances define the space shown in Figure 5, to which variational calculus with mobile frontiers may be applied (reuse stage) to calculate the optimum solution.

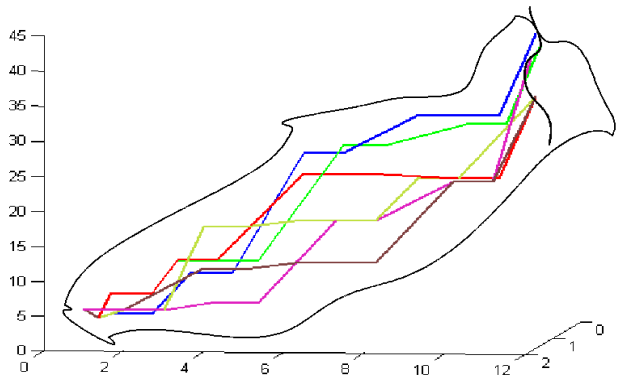

Figure 5: Surface or space to which variational calculus with mobile frontiers may be applied.

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 6 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.

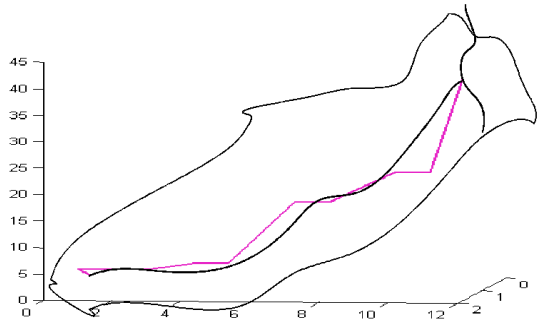

Figure 6: Optimum and closest to the optimum route.

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. Again, the new routes may be represented, 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.**

5. Results and conclusions

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. 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.

Acknowledgements

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

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] 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.

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

[4] 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

[5] 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.

[6] 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.