Computing and Information Systems **9** (2002) p. 2-13

A planning strategy based on variational calculus for deliberative agents

M. González Bedia and J. M. Corchado

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 introduced in this paper to facilitate to the agents the planning and replanning of their intentions in 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.

1 INTRODUCTION

Technological evolution in today's world is fast and constant. Successful systems should have the capacity to adapt at 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 (Wooldridge and Jennings, 1995). This paper shows how a deliberative agent with a BDI (Believe, 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 in execution time.

Agents should be autonomous, reactive, pro-active, sociable and have learning capacity. They must be able to answer 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 present goals. There are different types of agents and they can be classified in different ways (Wooldridge and Jennings, 1995). 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 get it (Wooldridge and Jennings, 1995; Jennings, 1992): 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 attain; 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 (Bratman, 1987; Kinny and Georgeff, 1991). 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 (Cohen and Levesque, 1990; Jennings, 1992; Shoham, 1993). 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 (1995) assert that the problem lies in the great difference between the powerful logic of BDI systems and with that required by practical systems. Another problem is that these types of agents don't have learning capacity - 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 formalisation 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 facilitate 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, intention but also with learning capacity.

Although the relationship between agents and CBR systems have been investigated by other researchers (Martin *et al.*, 1999; Wendler and Lenz, 1998; Olivia *et al.*, 1999), 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 retrieval stage of the CBR cycle to guaranty an efficient planning and replannig in execution time. Although different types of planning mechanism can be found in the literature (Camacho *et al.*, 2001; Knobolock et al., 2001; Laza and Corchado, 2001), none of them allow the replaning in execution time, and agents inhabit changing environment in which replaning in execution time is required if goals have 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 the changes on the environment. For instance, in Knobolock et al., 2001; Laza and Corchado, 2001 it is introduces a kind of plan schemas that need to be reprogrammed overtime, when the planning domain changes. In Camacho *et al.*, (2001) it is proposed an architecture that tries to be more flexible by using planning strategies to create the plans. If new information must to be introduced from the environment to the system, it is only necessary to change the planning domain instead of reprogramming the plan schema by hand. This architecture allows building plans that contain steps with no detailed information. This is useful because if no specific information is supplied, the solution can handle planning generic operators, plans that are not influenced by unexpected changes. Now to know if the abstract proposed plan is adequate it is required 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 it is proposed a solution that deals adequately with environmental real-time problem changes without applying a reprogramming strategy and without the disadvantages shown in Camacho *et al.*, (2001) because the technique used can solve a planning problem in 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 system and deliberative agent 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 will be shown how it can be used to define agents with the previously mentioned characteristics. Finally it is shown how it is possible to define an agent for the e-tourism domain using the methodology presented, together with the conclusions.

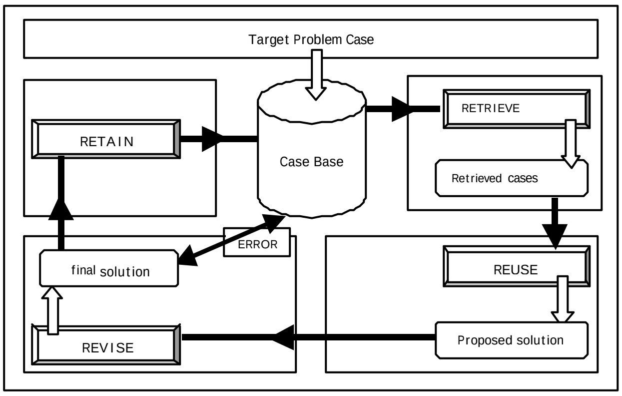

Figure 1: CBR Cycle of Life.

2 CASE-BASED REASONING SYSTEMS

Case-based reasoning is used to solve new problems by adapting solutions that were used to solve previous similar problems (Corchado and Lees, 2001). 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 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 (Aamodt and Plaza, 1994; Watson and Marir, 1994). 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 (Corchado and Lees, 2001; Fyfe and Corchado, 2001). 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 (Martin *et al.*, 1999; Wendler and Lenz, 1998; Olivia *et al.*, 1999), 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 the planning and replanning in execution time. The notation used in the refereed works (Martin *et al.*, 1999; Wendler and Lenz, 1998; Olivia *et al.*, 1999), 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:

Θ is the set that describes the agent environment.

$T(\Theta)$ is the set of attributes $\{\tau_1, \tau_2, \ldots, \tau_n\}$ in which the world is expressed.

- *Definition. A* ***belief*** *e on* Θ *is a m-upla of attributes of* $T(\Theta)$

  $e = (t1, t2, \ldots, tm)$ *with* $m \le n$

- *Definition. We call set of beliefs on* Θ *and denote* $?(\Theta)$

  $?(\Theta) = \{ (t1, t2, \ldots, tj) \text{ where } j = 1,2,\ldots, m \le n \}$

- *Definition. Let us introduce the operator ? of accessibility between two beliefs (e1, e2) and denote* $\Lambda(e1, e2) = e1 \wedge e2$, *that joins beliefs if they are compatible.*

- *Definition. Let us say that two beliefs are not accessible if ? (e1, e2)=0.*

- *Definition. An intention i on* Θ *is a s-upla of beliefs compatible to each other*

  $i = (e1, e2, \ldots, es)$ *with s ? IN and ? (ei, ej)* $\neq 0$

- *Definition. We call set of* ***intentions*** *on* Θ *and denote* $I(\Theta)$

  $I(\Theta) = \{ (e1, e2, \ldots, ek) \text{ where } k\ ?\ IN \}$

Now a set of parameters will be associated to the space $I(\Theta)$ that characterises any element of that set. The set of necessary and sufficient variables to describe the system may be obtained experimentally.

- *Definition. We call canonical variables of a set* $I(\Theta)$ to any set of *linearly independent* parameters $\aleph = (A_1, A_2, \ldots, A_v)$ *that characterise the elements i?* $I(\Theta)$.

- *Definition. A* ***desire*** *d on* Θ *is a mapping between*

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

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

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

A desire d may be achieved developing a plan – constructing an intention i – using some of the available beliefs, whose output could be evaluated in terms of the desired goals.

- *Definition. We call set of desires on* Θ *and denote* $D(\Theta)$

  $D(\Theta) = \{ d: I(\Theta) \rightarrow \Omega(\aleph)$ */ where* $I(\Theta)$ *is the set of intentions and* $\Omega(\aleph)$ *the set of mappings on* $\aleph$ *}*

Now, after presenting our definition of belief, desire and intention, section 3.2 defines the components of a CBR system.

3.2 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 $P = (S_i, \{ ?_j \}, S_f)$ with S_i=initial state, S_f=final state and $j = 1, \ldots, m$

S(P) is defined as $S(P) = \{S_k, ?_l\}$ where $k = 1, \ldots, n+1$ and $l = 1, \ldots, n \le m$, $S_1 = S_I$ and $S_{n+1} = S_F$

i.e. $S(P) = \{ S_1, ?_1, S_2, ?_2, \ldots, ?_n, S_{n+1} \}$

The state S_k and the operator $?_j$ are defined as

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

$$\text{where} \begin{Bmatrix} \{O_r\}_{r=1,\ldots,p} \\ \{R_s\}_{s=1,\ldots,q} \end{Bmatrix}$$

are coordinates in which a state S_k is expressed

- *Definition. We introduce the coordinates type $\{O_r\}_{r=1,\ldots,p}$ to express the objectives held.*
- *Definition. We introduce the coordinates type $\{R_s\}_{s=1,\ldots,q}$ to express the resources lost.*

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

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

The definition implies that $0 \le \Im_x \le 1$ and $0 \le \Im_y \le 1$.

In order to evaluate the rate of objectives achieved and resources used, between S and S', it is necessary to normalise every component of $\{O_r\}_{r=1,\ldots,p}$, $\{R_s\}_{s=1,\ldots,q}$

If $\{O_r(S)\} = (O_1, O_2, \ldots, O_P)$ and

$\{O_r(S')\} = (O'_1, O'_2, \ldots, O'_P)$

$\{R_s(S)\} = (R_1, R_2, \ldots, R_q)$ and

$\{R_s(S')\} = (R'_1, R'_2, \ldots, R'_q)$ it is defined

$$\Im_x = \frac{\sqrt{\left(\frac{O_1-O_1}{\max O_1}\right)^2 + \left(\frac{O_2-O_2}{\max O_2}\right)^2 + \ldots + \left(\frac{O_p-O_p}{\max O_p}\right)^2}}{\sqrt{\left(\frac{\max O_1}{\max O_1}\right)^2 + \left(\frac{\max O_2}{\max O_2}\right)^2 + \ldots + \left(\frac{\max O_p}{\max O_p}\right)^2}} = \frac{\sqrt{\left(\frac{O_1-O_1}{\max O_1}\right)^2 + \left(\frac{O_2-O_2}{\max O_2}\right)^2 + \ldots + \left(\frac{O_p-O_p}{\max O_p}\right)^2}}{\sqrt{p}}$$

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

A new parameter is also introduced - efficiency - that measures how many resources are needed to achieve an objective.

- *Definition. Given a target problem P, and a solution S(P), let us say that $\varsigma[S(P)]$ is the* efficiency *of the solution S(P) where*

  $$\varsigma[S(P)] = \Im_x / \Im_y$$

  The definition implies that $\varsigma(S,S') \in (0, \infty)$

- *Definition. A case C is a 3-upla {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.*
- *Definition. A CBR´s case base CB, **CB = {C_k / k=1,..,q and q ? IN}** is a finite set of cases memorised by the system.*

3.2.1 CBR life cycle

The operations that are carried out during the reasoning process of the CBR system are now defined, using the previously introduced notation. First, a mapping is introduced that associates an index idx(Ck) to a given case Ck.

idx: CB → I(BC)

C → idx(C)= [(α1, a1), (α2, a2),..., (αm, am)] ,

αj, ? ?(BC) ,al ? IR, m ? IN

where the set I(BC) is the set of indices of a case base CB and, as shown, I(BC) is represented by frames composed of conjunction of attributes (α1, α2, ..., αm) ∈ T(BC) - the set of attributes of a CBR - and a_1, a_2,...., a_m which denote values of the domain.

Using the previously shown coordinates $\{O_r\}$, $\{R_s\}$ it can be obtained

idx:CB→ I(BC)

C → idx(C)= idx{P, S(P), $\Im$[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? ?(BC) ,a_i, b_j, c_k, d_l ? IR and p,q ? IN

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

  (CB, R)={ [C_k/ k=1,..,q and q ? IN], R}=

  = { (C_1, .., C_q)/ idx(C_1) ⊆ idx(C_2) ⊆ . ⊆ idx(C_q) }

- *Definition. Let us say that two cases C=*{S_I, S_F, S(P), $\Im$[S(P)]}, *C'=*{S'_I, S'_F, S(P'), $\Im$[S(P')]}∈ *CB, fulfil the relation* idx(C)⊆idx(C') if idx(S_I)⊆idx(S'_I) and idx(S_F)⊇idx(S'_F)
- *Definition. Let us say that $S(p_j)$ is a possible CBR solution of the target P*

  $$\forall C_j = (P_j, S(P_j), \Im[S(P_j)]) / idx(C_j) \supseteq P$$

Let us denote the set of possible solutions to a target problem P, as

$\Gamma = \{C_k \in CB / idx(C_k) \supseteq P\}$, $\Gamma \subset CB$, $k_{=1,\ldots,m}$

Now the reasoning steps may be defined as follows:

Retrieval

During this phase, a problem P´ stored in the case base BC and that is similar to the target problem P is identified.

- *Definition. Given a problem P and given P'/ C_q= (P', S(P'), $\Im$[S(P')])$\in$ CB, let us say that P' is the most similar problem to P, and we denote P'$\approx$P, if the case C_q is a possible solution CBR and holds $idx(C_q) \supseteq \{idx(C_k)\ k_{=1,....,q-1,q+1,...,m}\}$*

Adaptation

During the adaptation phase, the system executes a derivational reasoning mechanism [1] that can be represented by the adaptation function A, which can be used to obtain a solution S(P) through S(P')

$$A : (BC) X\ \Sigma (P) \rightarrow C$$

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

with P$\in$ Σ(P) called set of problems

- *Definition. Let us name "solution cbr" of a problem P and let us denote S(P) to*

S(P)=A[S(P')] where P'$\approx$P and A(C_q ,P) is the adaptation function CBR to P.

The adaptation function in now defined explicitly: If P=(S_I, S_F) and it exists C_q={P',S(P'),$\Im$[S(P')]}$\in$ (BC) where P'=(S'_I, S'_F) with S(P')= ($S'_{I=}S_1$, $?'_1$, S'_2, $?'_2$,..., $?'_n$, $S'_{n+1=}S_F$) so that P'$\approx$P, then the adaptation function A[S(P')] is defined as

$$A[S(P')]=\{S_{I=}S_1,?'_1,S''_2,?'_2,S''_3,?'_3,...,?'_n, S''_{n+1=}S''_F\}$$

With this definition, the effectiveness parameter takes the form $\Im\{A[S(P')]\}=\Im(S_I,S''_F)$

Revise

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_q=(P', S(P'), $\Im$[S(P')]) is recovered, given the similarity between P and P', the adaptation step ensures a solution S(P)=A[S(P')] but does not guarantee that:

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

Then, if the relationship [*] is fulfilled, the revision process is complete.

Memorisation

The problem target and the characteristics of the adapted solution can be memorised as a new case to be reused in the future and is denoted by

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

3.3 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). The advantage of this approach is that a problem can be easily conceptualised in terms of agents and then easily implemented in the form of a CBR system. So once the beliefs, desires and intentions of an agent are identified, they can be mapped onto a CBR system. First a number of definitions are introduced.

- *Definition. We use state ? of an intentional process $\{e_1\wedge e_2\wedge \ \ \wedge es_{-1}\wedge e_s\}$ to describe any of the situations intermediate to the solution $\{e_1\wedge e_2\wedge \ \ \wedge er$, with $r\le s\}$ that admit a representation over $\aleph=(A_1, A_2, \ ,A_v)$ fulfilling accessibility relationships among their components $\{\exists\ \Lambda(ei, e_j), _{i, j= 1,..,r\ and\ i < j}\}$.*
- *Definition. We define a representation space for the set* I(Θ), *and we use state space ϑ, to describe an euclidean space that holds a canonical variable over each axis.*

  In space ϑ

  - each state is represented by a spot ?

    $?_0= (A_1, A_2,..,A_v)_0 = < a1_0, a2_0,..., av_0 >$

  - each process between states is represented by a contour line ?

    $$? : \vartheta \longrightarrow \vartheta$$

    $$(e_1\wedge ..\wedge er) \rightarrow (e_1\wedge ..\wedge e_r\wedge e_{r+1})\ \text{if}\ \exists\ \Lambda(e_r, e_{r+1})$$

  - each constraining relationship among the variables $(A_1, A_2,..,A_v)$ is represented by a surface $p(A_1, ... , A_v)=0$
- *Definition. Given a canonical coordenate system $(A_1, A_2,..,A_v)$ in* I(Θ), *the set may be reordered, differentiating between $\{F_m\}= \{A_j\ _{with\ j\ \le v}\ / A_j$ growing} and $\{G_n\}= \{A_k\ _{with\ k\ \le v} / A_k$ decreasing}, so $\aleph = \{F_m\}\cup\{G_n\}$ and m+n=v*
- *Definition. Giving an i$\in$* I(Θ), *a functional dependence relationship may be obtained in terms of the attributes*

$$i = i\,[e_1(t_1, t_2,\ldots,t_j), e_2(t_1, t_2,\ldots,t_k),\ldots,e_s(t_1, t_2,\ldots,t_q)]$$

$$= i\,(t_1, t_2,\ldots,t_n)$$

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.

The solution S(P) for a given problem $P=(S_I,\{?_j\},S_F)$ can be seen as a sequence of states $S_k=(\{O_r\}_{r=1, ,p}, (\{R_s\}_{s=1, ,q})$ interrelated by operators $\{S_k, ?_l\}$.

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 $\aleph = (F_1, F_2,\ldots,F_m, G_1, G_2,\ldots,G_n)$.

If we establish the relationship between the set of parameters

$\{F_m\}$	______	$\{O_r\}$
$\{G_n\}$		$\{R_s\}$

an identification criteria among the interrelated states, $?_i \in I(\Theta)$, and the CBR states, $S_k \in T(BC)$, may be established, and therefore a relationship may be established among the agents desires $D(\Theta)$ 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 desire to achieve. Cases are then adapted to generate y proposed solution, which is the agent action plan. This initial solution is reviewed and finally a learning process is carried out by adapting, deleting, etc. cases.

The following section shows how the retrieval stage can by automated using variational calculus (Arnold, 1971; De Groot, 1970), which facilitate the agents planing and replaning in execution-time.

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 (Corchado and Lees, 2001; Fyfe and Corchado, 2001) depending on the problem to solve. This section shows how variational calculus can be used during the retrieval stage of such agents to facilitate the planning and replanning of their intentions in execution-time.

Variational calculus is therefore used in the framework of the CBR system to automate the retrieval stage, which gives the agents more autonomy. In general variational calculus provides the optimum solution (geodesical) to a problem (Morse and Feshbach, 1953). Since we are using this mathematical formalism in a discrete environment (*cases: believes, desires and intentions*), it will be used to obtain the closest discrete solution to the optimal one (Morse and Feshbach, 1953).

Agents are dynamic systems, which should be able to respond to changes on real time, to plan their solutions and to modify such plans if the changes on the environment require it. When a CBR system is integrated within a BDI-agent structure, the changes on the environment introduce new criteria to identify a satisfactory CBR solution. This analytical formalism provides the necessary tools for the definition of a dynamic CBR-BDI agent, capable of planing and replaning (identification of intentions) in execution-time.

4.1 Mathematical foundations: variational problems

Suppose a space m-dimensional ?=(X1, X2,...,Xm) and a mapping on X, V(?), that is defined as

$$V : ? \longrightarrow ?$$

$$(X1, X2,...,Xm) \longrightarrow V\,(X1, X2,...,Xm)$$

On the phase space, which is the set of all states of the process, the function V(X1, X2,...,Xm) becomes an m-1dimensional surface that shows all possible relationships between the parameters of V, that is denoted G(X1, X2,...,Xm) = 0

- *Definition. Let us consider two points, ei and ef, that fulfil that:* $e_i = (X_i1, X_i2,...,X_im)$ *holds* $G(e_i)=0$ *and* $e_f = (X_f1, X_f2,...,X_fm)$ *holds* $G(e_f)=0$. *It is defined the set* $f(e_i,e_f)=\{f_1(e_i,e_f), f_2(e_i,e_f),., f_m(e_i,e_f)\}$ *where* $f_j(e_i,e_f)$ *are possible curves between* e_i *and* e_f *that are allowed by V(X1,X2,...,Xm) on ?=(X1, X2,...,Xm)) and hold that*

  $\forall\, f_j(X1, X2,...,Xm) \in \{ f(e_i,e_f) \}$ *it is satisfied* $G[f_j(X1, X2,...,Xm)]=0$

Given the set $f(e_i,e_f)$, variational calculus shows how the optimal curve (geodesical) with respect to its length can be chosen automatically (Arnold, 1971; De Groot, 1970; Morse and Feshbach, 1953).

If m=3 (see Figure 2) and we denote X = (X1, X2, X3) = (X, Y, Z), the following definition may be included.

- *Definition. A functional A[y=y(x)] defined on a space F is a continuos mapping of F into real numbers*

$$A:\ \Omega^{\infty}(IR) \longrightarrow IR,$$

$$y=y(x) \longrightarrow A[y=y(x)]$$

where Ω^{∞} *(IR) is the set of functions infinitely differentiated on IR*

If we have a functional A, we can demonstrate that the extreme solutions to this functional are the functions $y=y_0(x)$ *such that dA[y(x)] =0*

- *Definition. A functional A[y(x)] is called integrable if its expression takes the form*

$$A\,[y(x)] = \int_{x1}^{x2} [\, F(x, y(x), y'(x)\,]dx$$

For these cases, it is known that a function $y=y_0(x)$ *is optimal for the functional A[y(x)] if the Euler´s equation is satisfied* [2].

A[y(x)] extremal ¬ ® dA[y(x)] =0 ¬ ®

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0 \quad ¬ ® \quad y = y_0(x)$$

- *Definition. Let us define on the surface G(X,Y,Z)=0 generated from V=V (X, Y, Z), the notion of Euclidean distance that associates to each pair of points (ei, ef) a real number D(ei , ef) obtained as*

$D(e_i, e_f) = \sqrt{(X_i - X_f)^2 + (Y_i - Y_f)^2 + (Z_i - Z_f)^2}$ *where* $e_i = (X_i, Y_i, Z_i)$, $e_f = (X_f, Y_f, Z_f)$

So the length of a curve is given by

$$L = \int_{x1}^{x2} [dl] = \int_{x1}^{x2} \sqrt{dx^2 + dy^2 + dz^2} =$$

$$= \int_{x1}^{x2} \sqrt{1 + y'(x)^2 + z'(x)^2}\, dx$$

It is known that G(X, Y, Z)=0 implies constraints between y(x), z(x), and it defines a pair of new coordinates (?,?) that yields a new equation to be solved,

$$L = \int_{q1}^{q2} \sqrt{1 + r(q)^2}\, dq \quad = L\ \{\ ?, ?(?), ?'(?)\ \}$$

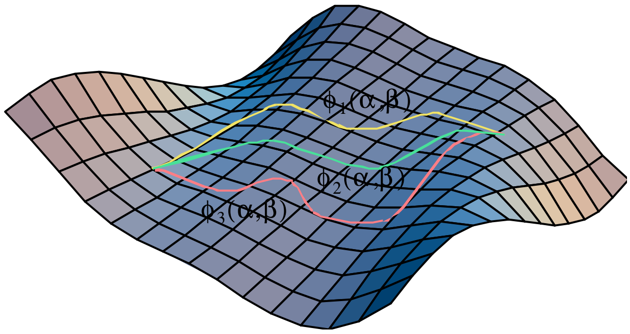

Figure 2: Possible paths between two points in a 3D space.

an expression in which Euler´s equation may be applied because L, with the previously shown dependence, is an integrable functional. A generalisation of Euler´s equation exists valid for any number of parameters. In this case, the solution is obtained solving an n-dimensional Euler´s system of differential equations.

4.2 Planning with variational calculus

This section shows how the variational calculus, introduced in the previous section, allows the agents planning and replanning in execution-time because this formalism is used to select the most adequate case during the retrieval phase of the reasoning process to solve a given problem. This mathematical formalism guaranties that the agent plan is the closest to the optimum (geodesical), it can not be the optimum since we are working with a discrete environment. This section also outlines how Green's functions may be used to improve the planning incorporating updated knowledge about the evolution of the environmental parameters that defines the problem. Figure 3 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 it can also be appreciated graphically the working and information flow during the reasoning process of the agent, which was introduced in section 3.2.1 and 3.3.

Let us consider a case base in a CBR, **(CB, R)={[C_k / k=1,.,q and q ? IN],R}** and the set of attributes of the case base ?(BC)= ($\alpha_1, \alpha_2,..., \alpha_m$), α_j, ? ?. Using the relationships between BDI agents and CBR systems established in section 3.3. it is denoted ?(BC)=(A_1,A_2,...A_v), where {A_j} is a canonical coordinates system of I(Θ), which allows us to define a function V on the space I(Θ), taking the form V=V(A_1,A_2,...A_v) that stores the information of all the cases Ck **? CB**.

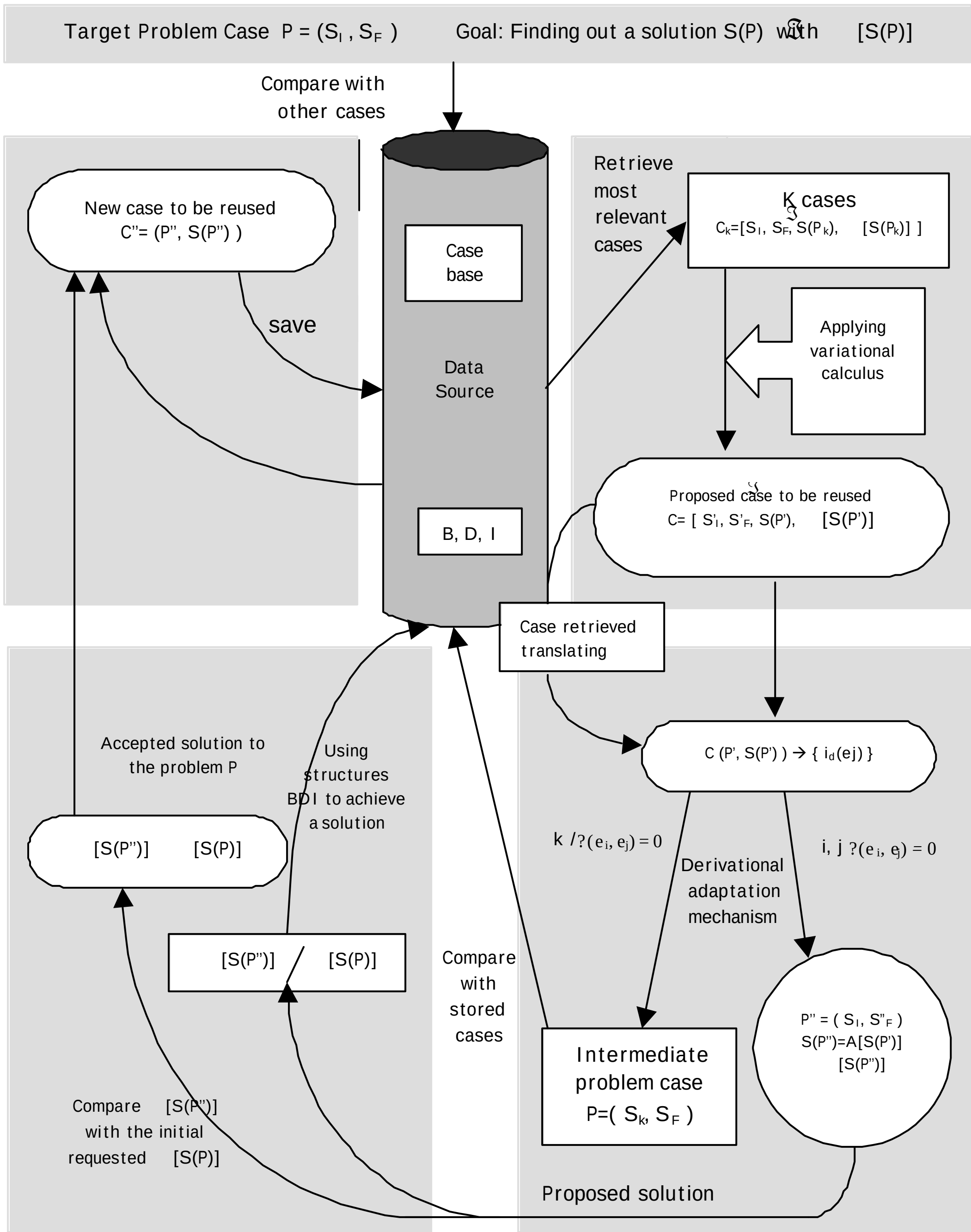

Figure 3: Formal Model Detailed Schema

If we consider two states (Si , Sf) initial and final, on the set I(Θ), the function V shows all the intentions i ? I(Θ), that joins both states (Si ,Sf) and that has related a case C_k ? CB.

On the phase space, the function **V=V(**$A_1, A_2, ... A_v$**)** is translated onto a surface **?** $_0$[$A_1, A_2, ... A_v$]**=0,** where the notion of Euclidean distance is defined.

Let S_i , S_f be two points, 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 **(?, µ)** exists which allows an expression of the functional **F=F(?, µ),** that associates to each curve between S_i and S_f on **?** $_0$**[x,y,z]=0** with its length, thus we can obtain a solution of

$$\frac{\partial F}{\partial m} - \frac{d}{dl}\left(\frac{\partial F}{\partial m'}\right) = 0$$

that we call **µ = µ** $_0$ **(?)**

and that takes the form **?**$_0$ **= ?**$_0$**[x,y,z]** on the original

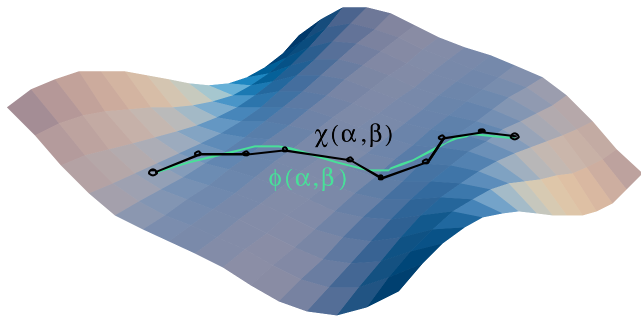

Figure 4: Optimum (geodesical) and closer to the optimum curve.

coordinates (X, Y, Z). This function is named the geodesical curve.

In the most general case, the mapping $V=V(A_1,A_2,\ldots,A_m)$ generates curves that cannot be differentiated because **V** only takes values at discrete points corresponding to defined and stored cases.

- *Definition. Let us now define a mapping* ѕ, *as* ***s****= ?$_0$ – ?, where ?$_0$ is the solution obtained by Euler´s equation (geodesical) [14] and ? ? {f(S$_i$,S$_f$)} is a path between Si, and Sf, stored in the case-base as a case,e C ? CB (see Figure 4).*
- *Definition. Let us call the closest to the optimal curve ?$_0$ the mapping of* ***{****f (S$_i$,S$_f$)****}*** *given by the minimisation of*

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

written as **? $_0$ = { 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 }**, where S_k are 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.

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 (Jennings, 1992).

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. Nevertheless the reasoning process may be maintain since the problem general description 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 **? $_1$ (X, Y, Z) = 0** on the phases space**,** upon which it is possible to obtain the optimal curve between two new points, S_i and S_f where

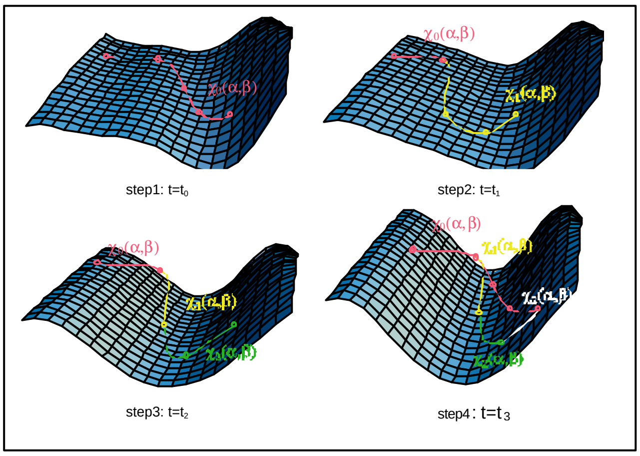

Figure 5: 3D representation of a dynamic environment.

S_i = $S_1^{(0)}$, and **$S_1^{(0)}$** is the second state of **? $_0$ = { S_i = $S_0^{(0)}$, $S_1^{(0)}$,....., $S_s^{(0)}$ = S_f }**

S_f is the final state or solution state of the global problem.

Solving the Euler´s equations, **?$_1$ = ?$_1$(X, Y, Z**) is obtained, which may be used to calculate an expression *for* **?** $_1$, through the *mapping* ѕ

? $_1$ = { S_i = $S_1^{(0)}$, $S_1^{(1)}$, $S_2^{(1)}$, $S_3^{(1)}$, $S_4^{(1)}$,...., $S_m^{(1)}$ ......, $S_s^{(1)}$= S_f }

and the same can be done for any **t_j** (See figure 5).

- *Definition. 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:*

$$?_{final} = \begin{cases} ?_{0}, & t\,?\,(t_0, t_1) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ ?_{s-1}, & t\,?\,(t_{s-2}, t_{s-1}) \\ ?_{s}, & t\,?\,(t_{s-1}, t_s) \end{cases}$$

Green´s functions may be used for the automation of the planning process. If there exist knowledge about the evolution of the problem in advance, Green´s functions theory can be applied to obtain a functional form of a time-dependent solution (Morse and Feshbach, 1953).

With the mappings obtained overtime:

V_0 (X, Y, Z), V_1 (X, Y, Z), V_2 (X, Y, Z), , V_r (X, Y, Z) for **t_0, t_1, t_2, t_3, ….t_r**

the following definitions may be obtained,

- *Definition. Let us define the time-dependent generating function V=V[*X, Y, Z, *t] that associates to each tj a Vj [*X, Y, Z*].*

For instance, in t_0, the function V_0 [X, Y, Z] takes the form V [X, Y, Z, t_0], which gives the solution **$?_0$=$?_0$[X, Y, Z] = ?[X, Y, Z, t_0],** after the Euler equation are solved.

- *Definition. Let us define the propagator of a function V [*X, Y, Z*, t] and we denote*

  G(?, ?', t, t_0) where ?=[x,y,z] *?'*= [x',y',z'] and

$$G (?, ?', t, t_0) =\int ? [?', t_0] V(?, t) ? [?', t_0] dx'$$

The propagation of a function from the time $\mathbf{t_0}$ to the time **t** is controlled by the function G(?,?', t, t_0), which we have called propagator. Such propagator is also known as Green´s function and stores all the information concerning the dynamic of a problem. Thus, we conclude this section by mentioning that theGreen´s functions theory, can be used to define the best plan (agent-intention) in order to fulfill the agent's desire.

$$?[?, t]= \int G (? , ?', t, t_0) ?[?', t_0] dx$$

5 CASE STUDY: CBR-BDI AGENTS IN THE TOURISM ENVIRONMENT

The framework in which this mathematical formalisation and experiments are being developed aims to design and implement an agent-based tool, as well as integrating existing state of the art in order to create an open, flexible, global anticipatory system with mobile access for the promotion and management of inland and cultural tourism, which will be user-friendly, cost-effective and secure. The system will be standardised and interlingual. It will be aimed as both a B2B and B2C tool and thereby help individuals, private enterprise and public bodies connected directly and indirectly to tourism to achieve higher quality of service.

The integrated, multi-platform computer tool developed has been specifically designed for the promotion of inland and cultural sites for tourism based on their cultural worth, the recreational activities on offer and new perspectives on sources of patrimonial interest. This will be combined with horizontally and vertically compiled information on hotel accommodation, restaurants, the commercial sector and transport, in order to meet the needs of the potential visitor on an individually customised basis and respond to requests for information, reservations and purchases in the precise moment that they are expressed.

The project aims to develop innovative, practical and multidisciplinary solutions which aim to use the varied knowledge of individuals at each location, and to organise the different services – often offered chaotically by different sources – within a single, dynamic, interconnected knowledge system. In order to achieve this, it will be necessary to integrate within a TIC platform, information that will facilitate the management of the knowledge lifecycle of various organisations.

One of the initial steps in this research is to develop an agent architecture for modelling autonomous agents. In this context our first experiment has been to design an individual agent, using the previously presented formalisation, whose aim is to assist tourists in identifying an optimum schedule for a day trip in the city of Salamanca. The users of the system interact with the agent via Internet or Mobile devices. The tourists select a number of items to visit, the type of restaurant that they prefer and the amount of money that they have to spend. Then the agent proposes an optimum schedule and if the tourist changes this plan on the move, that agent will (replan in execution-time) propose a new schedule.

The beliefs, desires and intentions of the agent are stored in the form of cases. In this problem, cases are solutions to previous tourist requests. The agent's goal is to select the plan that fulfill the requirements of the tourist efficiently. The routes are selected depending on the tourist criteria, for example, a criteria may be:

- visit the maximum amount of places (P)
- minimum cost (C) (tourists want to save as much money as possible)
- minimum time (T) (tourists don't want to waist their time)

These criteria are used to identify a requested route and are the elements that the agent takes into consideration to achieve its goal. The tourists decide upon the values of (P, C, T) in order to say when a solution is adequate. They choose how much resources they are prepared to spend, and how many objectives will be accepted. This information is introduced through the effectiveness parameters, presented in section 3.2. The agent retrieves the cases from the case base that may be used to generate the agent's plan and variational calculus is used to build the intention (plan) that fulfil its goal.

	ATTRIBUTE	VALUE
i	IDENTIFICADOR	Iden-1
cl	CLASS	MONUMENT
s	TIME OPEN MORNING	10-12 hrs.
	TIME OPEN AFTERNOON	17-20 hrs.
c	COSTS (MORNING)	6 €
	COSTS (AFTERNOON)	12 €
h	TIME MEDIUM OF VISIT	1 hr.
z	ZONE OF PLACE	A
q	QUALITY INDEX	1

Table 1: Item attributes.

	ATTRIBUTE	VALUE
i^2	IDENTIFICADOR	Iden-A
Cl	CLASS	TRAVEL BY TAXI
Z^2	DISTANCE	D(A,A)
h	TIME	1 hr.
c	COSTS	12 €
q	INDEX QUALITY	1

Table 2: Distances between items

Table 1 shows the attributes that describe a belief, in this case are related to a monument, and that may be related to elements such as quality (q), economic costs (e), geographical place (z), opening times (h)…

Table 2 shows other types of possible belief, such as the distances between city areas, which are characterised by other attributes such as specific area(z^2), time of travel (h),…

With the previously introduced mathematical notation, the beliefs shown take the following form (refer to section 3.1)

e_1 =e (i, s, c, h, z, q) for the item shown in table 1, and

e_2 = e (i, z^2, h, c, q) for the item shown in the table 2.

Through the relationship between BDI agents and CBR systems (see section 3.2.), the set of attributes used to characterise the cases in this particular problem takes the form:

?(BC)=[(a_1 =P, a_2=T, a_3=C)], and the set of intentions that allow to achieve a given goal is expressed as ℵ =(A_1=P, A_2=T, A_3=C)where P: places visited, T: time of the visit, C: costs associated.

Then, it is possible to find a mathematical function that expresses every canonical variable (P, T, C) in terms of the attributes of $T(\Theta)$ (see section 3.1). In this case, the parameters T and C have an straight and explicit functional relationship. P requires a more complex expression.

$$P= P(s_j, h_k, z_l, z^2{}_m)$$

$$T=T(t_k) = \sum t_k, \text{ with } k=1,..,s$$

$$C=C(c_i) = \sum c_i, \text{ with } i=1,...,s$$

Using this framework, V is defined by V=V(P,T,C) on the set of attributes of the case base BC (see section 4.2) and it defines a surface **? $_0$ [P,T,C] = 0**, on the phases space, in which the notion of Euclidean distance is defined as

$$D(S_I, S_F) = \sqrt{(P_f - P_i)^2 + (T_f - T_i)^2 + (C_f - C_i)^2}$$

where S_I= (P_i, T_i, C_i) is the initial state, and S_F=(P_f,T_f,C_f) is a solution state defined by the parameters of effectiveness (refer to section 3.2).

The theory of variational calculus gives us a coordinate system **(** g**,** f **)** and associates to each route between two states, S_I and S_F, a solution of Euler´s equation which defines the solution to the given problem that is denoted by $f = f_0(g)$ (see section 4.1.).

In this case, the required solution is the route defined by the parameters of effectiveness.

The tourists may also change their schedule on the move for example if they decide to move the dinner time forward, repeat a visit to a particular place, etc. The CBR-BDI agent then will be able to adapt the plan to the new tourist requirements in execution-time. Also, since the CBR is learning continuously, the agent is learning too and could provide different schedules at different points in time for the same tourist query.

6 CONCLUSIONS

The CBR-BDI architecture solves one of the problems of the BDI (deliberative) architectures, which is the lacking of learning capacity. 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 formalisation 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 automatisation of the reasoning cycle of the agents, here presented.

Agents need to respond on real time to the user request and to adapt their solutions in real time, since the inhabit dynamic environments.. Variational calculus has been introduced in this paper to facilitate the agents to define their plans and to replaning in 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 plan is characterised for been the one that, in each of its stages, maintains most constant the efficiency ς [S(P)].

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 IST-2001-34240 project, by the CICYT projects TEL99-0335-C04-03 and SEC2000-0249 and the PGIDT00 project MAR30104PR.

Bibliography

Aamodt, A. and Plaza, E. (1994). "Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches", *AICOM*. 7(1): 39-59 MAR 1994

Arnold, V.I. (1971). *Ordinary Differential Equations*, Springer-Verlag, UK

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

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):385-390.

Cohen, P.R. and Levesque, H.J. (1990). "Intention is choice with commitment", *Artificial Intelligence*, 42 (2-3): 213-261 MAR 1990

Corchado, J.M. and Lees, B. (2001). "A Hybrid Case-based Model for Forecasting", *Applied Artificial Intelligence*, 15(2):105-127.

De Groot, M.H. (1970). *Optical Statiscal Decisions*. McGraw-Hill, New York.

Fyfe, C. and Corchado, J.M. (2001). "Automating the construction of CBR Systems using Kernel Methods", *International Journal of Intelligent Systems*, 16(4): 571-586

Jennings, N.R. (1992). "On Being Responsible", in *Decentralized A.I. 3*, (Y. Demazeau and E. Werner, ed.), North Holland, Amsterdam, The Netherlands.

Kinny, D. and Georgeff, M. (1991). "Commitment and effectiveness of situated agents", *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91)*, pp82-88, Sydney, Australia.

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

Laza, R and Corchado, J.M. (2001). "Creation of Deliberative Agents Using a CBR Model", *Computing and Information Systems Journal*, 8(2):33-39.

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*, 9(3):319-341.

Morse, P.M and Feshbach, H. (1953). *Methods of Theoretical Physics*, McGraw-Hill, New York.

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

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

Shoham, Y. (1993). "Agent-Oriented programming", *Artificial Intelligence*, 60(1):51-92.

Watson, I. and Marir, F. (1994). "Case-Based Reasoning: A Review", *The knowledge Engineering Review*, 9(3): 327-354

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.

Wooldridge M., Jennings Nr. (1995)" Intelligent Agents Theory And Practice" *Knowl Eng Rev* 10 (2):115-152.

M. González Bedia and J. M. Corchado Rodriguez are at the University of Salamanca, Spain