

Agentes Autónomos con Arquitectura CBR-BDI

Manuel G. Bedia, Juan Manuel Corchado

Departamento de Informática y Automática

Universidad de Salamanca

Plaza de la Merced s/n, 37008, Salamanca, España

e-mail: corchado@gugu.usal.es

Resumen

Este artículo introduce un formalismo matemático robusto para la definición y desarrollo de agentes deliberativos con un motor de razonamiento basado en casos. El concepto de agente deliberativo es introducido y el modelo de sistema basado en casos es descrito utilizando un formalismo analítico. Se introduce posteriormente el cálculo variacional para facilitar a los agentes herramientas de planificación y replanificación de sus intenciones de forma que puedan reaccionar a los cambios del entorno en tiempo real. El continuo desarrollo en la industria turística y la adaptación de procesos que, mediante las nuevas tecnologías, se está haciendo, nos sugiere la formalización de un agente que asista a potenciales turistas en la organización de sus vacaciones y sea capaz de modificar sus horarios a través de sistemas de comunicación móviles.

Palabras Clave: Razonamiento basado en casos, Cálculo Variacional, Agentes y planificación.

1. Introducción

La evolución tecnológica en el mundo actual es rápida y constante. Los sistemas que tendrán éxito deben tener capacidad para adaptarse y deben disponer de mecanismos que les permitan decidir qué hacer para alcanzar sus objetivos. Tales sistemas existen y se conocen como agentes autónomos o inteligentes [20]. Nos planteamos desarrollar un agente con arquitectura BDI que utilice un mecanismo de razonamiento basado en casos para generar sus planes. Fundamentaremos este planteamiento con un robusto soporte matemático basado en el cálculo variacional, que nos permitirá integrar las arquitecturas BDI con un sistema CBR y nos permitirá introducir un mecanismo para la planificación y re-planificación del agente en tiempo real. Los agentes deben ser autónomos, reactivos, pro-activos, sociables y con capacidad de aprendizaje. Deben ser capaces de responder a eventos que tengan lugar en su entorno, tomar iniciativa de acuerdo a sus objetivos,

interactuar con otros agentes (algunos de ellos pueden ser humanos) y usar experiencias pasadas para alcanzar sus objetivos.

Existen diferentes tipos de agentes que pueden clasificarse de modos distintos. Uno de los más importantes son los agentes deliberativos BDI, cuya arquitectura se asienta sobre tres pilares: creencias, deseos e intenciones, y el agente las utiliza con el fin de estructurar las decisiones que le lleven a alcanzar sus objetivos: sus creencias representan su estado informacional –lo que los agentes conocen acerca de ellos mismos y de su entorno– sus deseos son sus estados motivacionales –lo que intentarán alcanzar– y sus intenciones representan sus estados deliberativos. Estas actitudes intencionales determinan el comportamiento de los agentes. Las arquitecturas BDI tienen la ventaja de ser intuitivas –es relativamente fácil reconocer el proceso de decisión y el modo de ponerlo en práctica–. Por otro lado, su principal desventaja reside en determinar un mecanismo que pueda permitirnos su adecuada implementación.

En el momento actual, el formalismo y la implementación de agentes BDI constituye el trabajo de investigación de numerosos científicos [5, 8, 17].

Algunos de ellos critican los trabajos en formalización a partir de lógicas multimodales para la formalización y la construcción de dichos agentes, porque no es un marco completamente axiomatizado ni computacionalmente eficiente.

Rao et Georgeff [16] afirman que el problema se encuentra en la gran diferencia que existe entre la potencia de esas lógicas y lo que se requiere para la actuación práctica de tales sistemas. Otro problema es que estos tipos de agentes no tienen capacidad de aprendizaje –un elemento imprescindible si queremos que constantemente incorporen, modifiquen o eliminen creencias, deseos e intenciones –.

En este trabajo se presenta una formalización analítica y de carácter robusto, frente a las alternativas en lógica modal, que además contempla un modo de implementación eficiente. Se muestra cómo la estructura de agente BDI a la que se incorpora un motor de razonamiento basado en casos CBR puede acelerar sustancialmente el proceso de deliberación del BDI y resolver los problemas relacionados con la capacidad de aprendizaje de los agentes: implementar agentes en la forma de sistemas CBR facilita su aprendizaje y adaptación.

Si logramos establecer una correspondencia entre las actitudes mentales de los agentes y la información que un sistema basado en casos manipula, entonces tendremos una forma de desarrollar agentes no sólo con sus creencias, deseos e intenciones, sino también con capacidad de aprendizaje.

Existen resultados de investigación que aportan luz en la relación entre agentes y mecanismos de razonamiento CBR [13, 19, 15]. En nuestra propuesta, el formalismo matemático que la soporta facilita los procesos para la correcta y eficiente implementación de un agente en la forma de un sistema CBR. El cálculo variacional se introduce para automatizar el ciclo de razonamiento del agente, fundamentalmente durante la etapa de recuperación, para garantizar un mecanismo de planificación y re-planificación eficiente en tiempo real.

Aunque se pueden encontrar diferentes tipos de mecanismos planificadores en la lite-

ratura correspondiente [4, 11, 12], ninguno de ellos nos permite re-planificar en tiempo real, y cada vez que los cambios en el entorno rompen con las pretensiones del agente de alcanzar los objetivos a partir del plan que había iniciado, el sistema debe "desandar lo andado" para preparar un nuevo plan.

Algunos de los trabajos planteados usan, tal como indicábamos, técnicas de planificación que no pueden tratar con los cambios producidos en el entorno y plantear modificaciones en tiempo real, por ejemplo, los trabajos de [11, 12] introducen un tipo de esquemas de planificación que necesitan ser reprogramados cuando el ambiente cambia. En [4] se propone una arquitectura que intenta ser más flexible en la re-planificación, al incorporar nueva información al sistema proveniente del medio, pues sólo se cambia el dominio informacional sobre el que se diseña el plan en lugar de reprogramar el plan por completo, y se conservan las líneas de razonamiento general. Esta arquitectura nos permite construir las sucesivas fases de un plan sin especificar demasiada información y, por tanto, sin depender excesivamente del dominio ni de los inesperados cambios en él. Posteriormente, para conocer si el plan trazado es adecuado, se requiere ponerlo en práctica en dominio real. Esta operación requiere una gran cantidad de tiempo computacional y emplea grandes recursos porque necesita hacer dos veces el mismo trabajo, primero diseñando el plan en términos generales, y después comprobando que es adecuado al problema, proyectándolo al caso concreto. Con el modelo que proponemos, se plantea una solución que trate adecuadamente con problemas donde los cambios en el entorno no impliquen ni estrategias de reprogramación, ni las desventajas mostradas en [4], porque existe un criterio de decisión que, ante cambios contextuales, puede tomar decisiones en tiempo real. La clave del sistema está en aprovechar la potencia del cálculo variacional al utilizarse dentro del ciclo de vida CBR, para automatizar su fase de recuperación.

Comenzaremos revisando los conceptos de procesos CBR y de sistemas deliberativos e introduciremos un único esquema de representación para ambos. A partir de esta notación común, es sencillo mostrar cómo integraremos el mecanismo CBR para operar con las actitudes mentales de un agente deliberativo. Se introduce, entonces, el cálculo variacional y se muestra cómo automatizar el razonamiento de agentes con las características mencionadas.

Finalmente, se presenta el esquema de funcionamiento de un diseño de agente BDI-CBR, para el sector del turismo electrónico, usando la metodología que hemos introducido.

2. Sistemas de razonamiento basado en casos (CBR)

Los sistemas de razonamiento basado en casos se usan para resolver nuevos problemas mediante la adaptación de soluciones que fueron usadas en la resolución de problemas similares [6]. El proceso de trabajo de un sistema CBR consiste, fundamentalmente, en la adaptación de viejas soluciones para construir planes asociados a nuevas experiencias, usando casos ya validados para plantear nuevas propuestas, para formular nuevas soluciones o para razonar de manera análoga en situaciones similares.

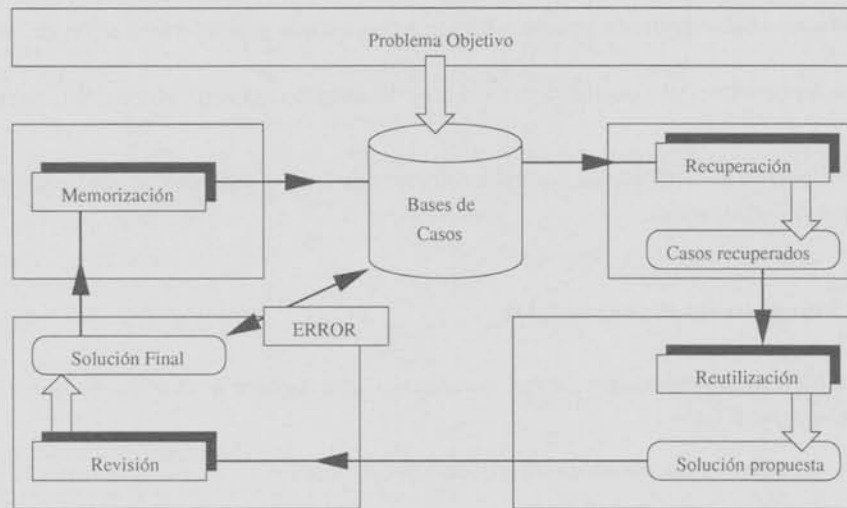


Figura 1: Ciclo de razonamiento del CBR.

La Figura 1 nos muestra el ciclo de razonamiento de un CBR típico, donde se incluye la secuencia de cuatro etapas que están cíclicamente conectadas: recuperación, reutilización, revisión y memorización [1, 18]. Durante la fase de recuperación, los casos que son más similares al problema que se nos propone se extraen de la base de casos. Los casos recuperados son adaptados para generar una posible solución durante la fase de reutilización o adaptación. La solución es entonces revisada y, si es apropiada, se le da forma y entidad de caso, para que en la fase de memorización se almacene en memoria.

Por tanto, los sistemas CBR actualizan constantemente su base de casos y en consecuencia se adaptan a los cambios del entorno. Cada una de las fases de razonamiento de un sistema CBR puede ser automatizada, lo que implica que podemos tener un modo automático para el proceso completo [6, 7]. Este resultado abre el camino a la hipótesis de utilizar sistemas CBR para implementar agentes BDI que pudieran razonar autónomamente y tener propiedades de adaptación a cambios en el entorno.

Los agentes pueden entonces usar el ciclo de razonamiento de un sistema CBR para generar planes. A partir de una notación que permita considerar casos y actitudes intencionales de manera común, se presentará un modelo para la implementación de BDI usando para ello un motor de razonamiento CBR.

3. Implementación de agentes deliberativos usando sistemas CBR

Esta sección presenta la relación que puede ser establecida entre agentes BDI y sistemas CBR y muestra cómo los agentes pueden razonar más eficazmente con la ayuda de un módulo CBR. La formalización presentada toma elementos de otros sistemas [13, 19, 15] y los adapta. Nuestro propósito es definir una relación entre agentes y su modelo de razonamiento, fijando especial atención en dos elementos:

1. Búsqueda de vertiente práctica teniendo en mente la implementación del modelo
2. Incorporación del aprendizaje mediante la adaptación a cambios del entorno

Para ilustrar el modelo, un ejemplo relacionado con el turismo es analizado paralelamente a la explicación.

3.1. Concepto de Agentes BDI

Introducimos la notación y la relación entre componentes actitudinales que caracterizan un agente BDI:

Θ es el conjunto que describe el entorno del agente

$T(\Theta)$ es el conjunto de atributos $\{\tau_1, \tau_2, \dots, \tau_n\}$ en el que el mundo es expresado

Definición. Una creencia e sobre Θ es una m -upla de atributos de $T(\Theta)$, que denotamos $e = (\tau_1, \tau_2, \dots, \tau_m)$ con $m \leq n$.

Definición. Llamamos conjunto de creencias sobre Θ y denotamos $\zeta(\Theta)$, al conjunto $\zeta(\Theta) = \{(\tau_1, \tau_2, \dots, \tau_j)$ donde $j = 1, 2, \dots, m \leq n\}$.

Ejemplo:

Supongamos que el mundo $T(\Theta)$ incluye todos los atributos necesarios para caracterizar una ruta turística en la ciudad de Salamanca (capital cultural europea 2002).

$$T(\Theta) = \{\tau_1 = \text{nombre monumento}, \tau_2 = \text{horario}, \tau_3 = \text{coste}, \\ \tau_4 = \text{tiempo de desplazamiento entre dos ítems de una ruta}, \dots, \tau_n\}$$

En este mundo, una creencia, por ejemplo, **monumento**, es un vector de atributos de $T(\Theta)$ que caracterizan el monumento.

$$\text{Creencia monumento} = (\tau_1 = \text{nombre}, \tau_2 = \text{horario}, \tau_3 = \text{coste visita}, \\ \tau_4 = \text{indicador de calidad})$$

Una creencia particular, por ejemplo, **catedral nueva**, es un vector como el anterior que valora en sus variables.

$$\text{catedral nueva} = (\text{nombre} = CN, \text{horario} \in (10:00h, 18:00h), \text{coste visita} = 3 \text{ €}, \\ \text{calidad} = 1.5)$$

Definición. Introducimos el operador “ Λ de accesibilidad” entre m creencias $(e_1, e_2, e_3, \dots, e_m)$, y denotamos $\Lambda(e_1, e_2, e_3, \dots, e_m) = (e_1 \wedge e_2 \wedge \dots \wedge e_m)$, al operador que crea nuevas estructuras uniendo creencias compatibles entre sí.

	Atributo	Valor
<i>i</i>	Entidad	<i>M1</i>
<i>cl</i>	Clase	Monumento
<i>s</i>	horario mañana	10-13 hrs.
<i>c</i>	tarifa mañana	6 €
<i>h</i>	tiempo medio visita	1 hr.
<i>z</i>	zona geográfica	A

(a) Valores de la creencia *M1*.

	Atributo	Valor
<i>i</i>	Entidad	<i>M2</i>
<i>cl</i>	Clase	Monumento
<i>s</i>	horario mañana	10-14 hrs.
<i>c</i>	tarifa mañana	6 €
<i>h</i>	tiempo medio visita	1 hr.
<i>z</i>	zona geográfica	A

(b) Valores de la creencia *M2*.

	Atributo	Valor
<i>i</i> ²	Entidad	<i>D(A, A)</i>
<i>cl</i>	Clase	Viaje en taxi
<i>h</i>	Tiempo medio viaje	1 hr.
<i>c</i>	Coste	12 €

(c) Valores de la creencia *D(A, A)*.

	Atributo	Valor
<i>i</i>	Entidad	<i>M3</i>
<i>cl</i>	Clase	Monumento
<i>s</i>	horario mañana	10-15 hrs.
<i>c</i>	tarifa mañana	6 €
<i>h</i>	tiempo medio visita	1 hr.
<i>z</i>	zona geográfica	A

(d) Valores de la creencia *M3*.

Tabla 1: Valores para las creencias *M1*, *M2*, *D(A, A)* y *M3*.

Definición. Diremos que el operador \wedge actuando sobre m creencias es nulo si alguna de ellas no es accesible, y lo denotamos $\wedge(e_1, e_2, e_3, \dots, e_m) = 0$.

Ejemplo:

Son las 12:00 pm, y el agente cree *M1*, *M2*, *D(A, A)*, creencias que se describen en las Tablas 1(a), 1(b) y 1(c).

Con estas creencias y dado que son las 12:00h, es imposible visitar *M1*, *M2*, por el camino ($M1 \wedge D(A, A) \wedge M2$) que, según nuestra notación, denotaremos $\wedge(M1, D(A, A), M2) = 0$.

Si *M2* es substituida por *M3* (ver Tabla 1(d)) entonces ($M1 \wedge D(A, A) \wedge M3$) es posible, y $\wedge(M1, D(A, A), M2) \neq 0$, lo cual significa que el agente ha identificado que podemos visitar los monumentos *M1* y *M3*, tomando en consideración que el tiempo de viaje entre ambos monumentos viene dado por *D(A, A)*.

Definición. Una intención *i* sobre Θ es una s -upla de creencias compatibles entre sí que denotamos $i = (e_1, e_2, \dots, e_s)$ con $s \in \mathbb{N}, \wedge(e_i, e_j) \neq 0$.

Definición. Llamaremos conjunto de intenciones sobre Θ y denotaremos $I(\Theta)$ al conjunto $I(\Theta) = \{(e_1, e_2, \dots, e_k) \text{ donde } k \in \mathbb{N}\}$.

Supongamos ahora que, a partir del análisis de un problema, determinamos un conjunto de parámetros que constituyen una base de atributos sobre la que estructurarlo. Se construye el conjunto \aleph , que toma los elementos de $I(\Theta)$ caracterizándolos a partir de

horario (hr)	10-11	11-12	12-13	13-14	14-16	16-18	18-20	20-21	21-22
intención	M1	D(A, A)	M2	D(A, A)	R1	D(A, B)	M3	D(B, B)	R2
Coste(€)	6	0	6	0	12	0	0	0	12
Tiempo (hr)	1	1	1	1	2	2	2	1	1
Calidad	1	-	2	-	1	-	2	-	2

(a) Valores de las creencias que constituyen la intención i_1 .

	Sistema de coordenadas de \aleph	
P	Número de monumentos	3
C	Coste(€)	36

(b) Valores para (P, C) asociados a i_1 .**Tabla 2:** Valores para las creencias de la intención i_1 y sus valores (P, C) asociados.

atributos de esta base. El conjunto necesario y suficiente de variables se obtiene empíricamente, ya que depende del problema propuesto y depende también en qué términos quiera resolverse.

Definición. Llamaremos “variables canónicas” de un conjunto $I(\Theta)$ a cualquier conjunto de parámetros linealmente independientes $\aleph = (A_1, A_2, \dots, A_v)$ que caractericen los elementos $i \in I(\Theta)$ –aunque llamamos variables canónicas, no son únicas, son propias de $I(\Theta)$, para cada problema P dado–.

Ejemplo:

Si un turista interactúa con el agente para solicitar una ruta que cumpla como requisitos, visitar un número de monumentos (P) y no gastar más que un coste máximo asociado (C), entonces podremos expresar $\aleph = (A_1, A_2) = (P, C)$.

En este sistema de coordenadas la intención siguiente:

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

con los valores de las creencias en la Tabla 2(a), tiene los valores de los parámetros P y C indicados en la Tabla 2(b).

Definición. Llamamos deseo d sobre Θ a una función entre

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

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

donde $\Omega(\aleph)$ es el conjunto de funciones sobre \aleph .

La expresión anterior nos dice que las actitudes intencionales que llamamos deseos pueden ser alcanzadas a través de un plan –que en nuestro esquema corresponde con la construcción de una intención adecuada– utilizando para ello, toda la información que

nos proporcionen las creencias, pero evaluadas en términos de los parámetros en los que se especifican los objetivos deseados.

Definición. Llamamos conjunto de deseos sobre Θ y denotamos $D(\Theta)$ al conjunto

$$D(\Theta) = \{d : I(\Theta) \rightarrow \Omega(\aleph) / \text{donde } I(\Theta) \text{ son las intenciones y } \Omega(\aleph) \text{ las funciones sobre } \aleph\} \tag{1}$$

Ejemplo:

La función deseo del turista “quiero una ruta en la que como mínimo vea tres monumentos y como máximo gaste 50€”, se puede expresar:

$$F(A_1, A_2) = F(P, C) = \left\{ \begin{array}{l} P \geq 3 \\ C \leq 50 \end{array} \text{ con } \begin{array}{l} P \in (0, 10) \\ C \in (0, 100) \end{array} \right\}$$

Hasta aquí hemos presentado el marco desde el que vamos a representar las actitudes básicas de un sistema BDI. Veamos, a continuación, cómo trabajamos con experiencias almacenadas en forma de casos y cómo razonar a partir de ellos.

3.2. Formalismo analítico para sistemas de razonamiento basado en casos

Consideremos un problema P , del cual deseamos obtener su solución que denotamos $S(P)$ El funcionamiento de un sistema basado en casos consiste en asignarle la solución $S(P)$ al problema, mediante la adaptación de una solución memorizada $S(P')$ asociada a un problema similar P' .

Denotamos P como $P = (S_i, \{\Theta_j\}, S_f)$ donde S_i = estado inicial, S_f = estado final, y $\{\Theta_j\}$ con $j = (1, \dots, m)$ es el conjunto de operadores a partir de los que se construye la solución. La solución $S(P)$ se define como $S(P) = \{S_k, \Theta_h\}$ donde $k = (1, \dots, n+1)$, $h = (1, \dots, n)$, $S_1 = S_i, S_{n+1} = S_f$, esto es, $S(P) = \{S_1, \Theta_1, S_2, \Theta_2, \dots, \Theta_n, S_{n+1}\}$.

Los estados S_k y los operadores Θ_j se definen como

$$S_k = \left(\begin{array}{l} \{O_r\}_{r=1, \dots, p} \\ \{R_s\}_{s=1, \dots, q} \end{array} \right), \quad \Theta_j : S_k = \left(\begin{array}{l} \{O_r\} \\ \{R_s\} \end{array} \right) \rightarrow \Theta_j(S_k) = \left(\begin{array}{l} \{O'_r\} \\ \{R'_s\} \end{array} \right) \tag{2}$$

donde $\{O_r\}_{r=1, \dots, p}$ son las variables en las que los estados del problema se representan. $\{R_s\}_{s=1, \dots, q}$

Definición. Llamaremos coordenadas tipo $\{O_r\}_{r=1, \dots, p}$ a aquellas que representan los objetivos de la solución.

Definición. Llamaremos coordenadas tipo $\{R_s\}_{s=1, \dots, q}$ a aquellas que representan los recursos que se van perdiendo en el proceso de solución.

A partir de las definiciones y la notación expuesta, introducimos el parámetro "eficacia, \mathfrak{F} " de la solución correspondiente a un problema $P = (S, S')$ que se define como un vector $\mathfrak{F}(S, S') = (\mathfrak{F}_x, \mathfrak{F}_y)$, con

$$\mathfrak{F}_x = \frac{O_r(S') - O_r(S)}{\text{máx}[O_r]} \quad \mathfrak{F}_y = \frac{R_s(S) - R_s(S')}{\text{máx}[R_s]} \quad (3)$$

esta definición implica $0 \leq \mathfrak{F}_x \leq 1$, $0 \leq \mathfrak{F}_y \leq 1$, y asocia para cada solución que se encuentre a un problema, un indicador con la información sobre objetivos alcanzados y recursos empleados que, al ser dividido entre el valor total, nos da idea del grado de satisfacción.

Con el fin de tener una expresión explícita que evalúe los componentes del vector eficacia entre los estados inicial y final, S y S' , de una solución $S(P)$, es necesario normalizar cada uno de los objetivos y recursos, para poder operar con valores, en principio heterogéneos, $\{O_r\}_{r=1, \dots, p}$, $\{R_s\}_{s=1, \dots, q}$.

Si $\{O_r(S)\} = (O_1, O_2, \dots, O_p)$ y en el estado final $\{O_r(S')\} = (O'_1, O'_2, \dots, O'_p)$

$$\mathfrak{F}_x = \frac{\sqrt{\left(\frac{O'_1 - O_1}{\text{máx } O_1}\right)^2 + \left(\frac{O'_2 - O_2}{\text{máx } O_2}\right)^2 + \dots + \left(\frac{O'_p - O_p}{\text{máx } O_p}\right)^2}}{\sqrt{\left(\frac{\text{máx } O_1}{\text{máx } O_1}\right)^2 + \left(\frac{\text{máx } O_2}{\text{máx } O_2}\right)^2 + \dots + \left(\frac{\text{máx } O_p}{\text{máx } O_p}\right)^2}} = \frac{\sqrt{\left(\frac{O'_1 - O_1}{\text{máx } O_1}\right)^2 + \left(\frac{O'_2 - O_2}{\text{máx } O_2}\right)^2 + \dots + \left(\frac{O'_p - O_p}{\text{máx } O_p}\right)^2}}{\sqrt{p}}$$

Como $\{R_s(S)\} = (R_1, R_2, \dots, R_q)$ y además $\{R_s(S')\} = (R'_1, R'_2, \dots, R'_q)$, tenemos,

$$\mathfrak{F}_y = \frac{\sqrt{\left(\frac{R'_1 - R_1}{\text{máx } R_1}\right)^2 + \left(\frac{R'_2 - R_2}{\text{máx } R_2}\right)^2 + \dots + \left(\frac{R'_q - R_q}{\text{máx } R_q}\right)^2}}{\sqrt{\left(\frac{\text{máx } R_1}{\text{máx } R_1}\right)^2 + \left(\frac{\text{máx } R_2}{\text{máx } R_2}\right)^2 + \dots + \left(\frac{\text{máx } R_q}{\text{máx } R_q}\right)^2}} = \frac{\sqrt{\left(\frac{R'_1 - R_1}{\text{máx } R_1}\right)^2 + \left(\frac{R'_2 - R_2}{\text{máx } R_2}\right)^2 + \dots + \left(\frac{R'_q - R_q}{\text{máx } R_q}\right)^2}}{\sqrt{q}}$$

Introducimos un nuevo parámetro, la eficiencia, que mide cuántos recursos se necesitan para alcanzar un objetivo.

Definición. Dado un problema P a resolver y una solución $S(P)$, diremos que $\zeta[S(P)]$ es la eficiencia de la solución $S(P)$ y se calcula como $\zeta[S(P)] = \mathfrak{F}_x / \mathfrak{F}_y$. La definición implica que $\zeta(S, S') \in (0, \infty)$.

Definición. Un caso C es una 3-upla $\{P, S(P), \mathfrak{F}[S(P)]\}$ donde P representa la descripción del problema, $S(P)$ la solución de P y $\mathfrak{F}[S(P)]$ la eficacia de la solución.

Definición. Una base de casos $CB = \{C_k / k = 1, \dots, q \text{ y } q \in \mathbb{N}\}$ es un conjunto finito de casos memorizados y accesibles por un sistema.

3.3. Integración de sistemas CBR en arquitecturas BDI

La relación entre sistemas CBR y agentes BDI puede ser establecida, asociando creencias, deseos e intenciones a casos. La ventaja de este enfoque es que un problema

puede ser fácilmente conceptualizado en términos de agentes y, posteriormente, implementado en forma de un sistema CBR. Primeramente, introducimos una función que asocia un índice $idx(Ck)$ a cualquier caso almacenado Ck . Si utilizamos la notación que presentamos para cada caso C , con coordenadas $\{O_r\}, \{R_s\}$, se tiene

$$\begin{aligned}
 idx : CB &\rightarrow I(CB) \\
 C &\rightarrow idx(C) = idx\{P, S(P), \mathfrak{F}[S(P)]\} = \{idx(S_i), idx(S_f)\} = \\
 &\quad \{[S_i = (O_1, a_1), (O_2, a_2), \dots, (O_p, a_p), (R_1, b_1), (R_2, b_2), \dots, (R_q, b_q)], \\
 &\quad [S_f = (O'_1, c_1), (O'_2, c_2), \dots, (O'_p, c_p), (R'_1, d_1), (R'_2, d_2), \dots, (R'_q, d_q)]\}
 \end{aligned}$$

donde $O_j, R_k \in T(CB), a_i, b_j, c_k, d_l \in \mathbb{R}, p, q \in \mathbb{N}$ y donde el $I(CB)$ es el conjunto de índices de una base de casos $CB(CaseBase)$ y, como hemos mostrado, se representan sus elementos por vectores compuestos por los atributos de $T(CB)$, y (a_1, a_2, \dots, a_m) , que comprende tanto valores individuales como dominios o intervalos de valores.

Definición. El proceso de abstracción que supone el mecanismo de indexación en el manejo de casos nos permite la introducción de una relación de orden R , en la base de casos CB , que nos permite comparar casos. Los índices están organizados en la forma de una jerarquía de subsumción.

$$(CB, R) = \{[C_k/k = 1, \dots, q \text{ donde } q \in \mathbb{N}], R\} = \{(C_1, \dots, C_q) / idx(C_1) \subseteq idx(C_2) \subseteq \dots \subseteq idx(C_q)\}$$

Definición. Diremos que dos casos $C, C' \in CB$, cumplen la relación

$$idx(C) \subseteq idx(C') \text{ si } \begin{cases} idx(S_i) \subseteq idx(S'_i) \\ idx(S_f) \supseteq idx(S'_f) \end{cases}$$

que expresado en componentes,

$$\begin{aligned}
 idx(S_i) \subseteq idx(S'_i) &\rightarrow \begin{cases} O_r(S_i) \geq O_r(S'_i) & \forall r = 1, \dots, p \\ R_s(S_i) \leq R_s(S'_i) & \forall s = 1, \dots, q \end{cases} \\
 idx(S_f) \supseteq idx(S'_f) &\rightarrow \begin{cases} O_r(S_f) \leq O_r(S'_f) & \forall r = 1, \dots, p \\ R_s(S_f) \geq R_s(S'_f) & \forall s = 1, \dots, q \end{cases}
 \end{aligned}$$

Definición. Diremos que $S(P')$ es una posible solución CBR de un problema P si $C' = (P', S(P'), \mathfrak{F}[S(P')])$ cumple la relación $idx(C') \supseteq idx(P)$, donde $idx(P)$ se calcula a través de las restricciones que a la solución buscada $S(P)$ se impone.

Ejemplo:

Dados dos casos, $C1, C2$, y sus índices, para los cuales, por sencillez, se supone que los estados iniciales son el origen de coordenadas y, por tanto, tienen valores nulos,

$$\begin{aligned}
 idx(C1) &= \{(O'_1, a_1), (R'_1, b_1), (R'_2, b_2)\} = \{(O'_1, 1.7), (R'_1, 95), (R'_2, 21.6)\} \\
 idx(C2) &= \{(O'_1, a_1), (R'_1, b_1), (R'_2, b_2)\} = \{(O'_1, 1.1), (R'_1, 80), (R'_2, 19.2)\}
 \end{aligned}$$

Con estos valores, una relación de subsunción no puede ser establecida. Si introducimos otro caso $C3$, con

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

Y si el problema a resolver puede ser representado por $P = (S_i, S_f)$, donde la solución buscada cumpla,

$$S(P) = \left\{ \begin{array}{l} O'_1 \geq 1 \\ R'_1 \leq 105, \quad R'_2 \leq 25 \end{array} \right\}$$

la relación $idx(C3) \subseteq P \subseteq idx(C1), idx(C2)$ puede ser establecida. Con las definiciones vistas, $idx(C3)$ no es "posible solución cbr de P ", mientras que $idx(C1), idx(C2)$ son "posibles soluciones cbr" para el problema.

Definición. Dada una base de casos CB , y dado un problema P , llamamos "conjunto de posibles soluciones cbr", al conjunto

$$\Gamma = \{C_k \in CB / idx(C_k) \supseteq P\}, \Gamma \subset CB, \quad k = 1, \dots, m$$

Con todo lo visto, tenemos el contexto preparado para introducir el concepto de CBR.

Definición. Llamaremos "estado ρ de un proceso intencional", que denotamos $\{e_1 \wedge e_2 \wedge \dots \wedge e_{s-1} \wedge e_s\}$, a la descripción de cualquier situación intermedia que el proceso cruce hacia la solución, $\{e_1 \wedge e_2 \wedge \dots \wedge e_r, \text{ donde } r \leq s\}$ expresada en términos de $\aleph = (A_1, A_2, \dots, A_v)$ y verificando que el operador accesibilidad aplicado a $\{e_1 \wedge e_2 \wedge \dots \wedge e_r, \text{ donde } r \leq s\}$ no es nulo, esto es $\{\exists \Lambda(e_i, e_j), i, j = 1, \dots, r; i < j\}$.

Definición. Construimos "un espacio de representación" del conjunto $I(\Theta)$, y denominamos "espacio de estados", denotado como ϑ , a un espacio euclídeo que tiene como base de coordenadas el conjunto de variables $\aleph = (A_1, A_2, \dots, A_v)$ –en el que se define una métrica que cumpla la propiedad euclídea– que preserve la noción intuitiva de distancia.

Estas definiciones preparan el marco donde exponer nuestra interpretación geométrica del razonamiento basado en casos, y permitirá enfocar el problema utilizando tratamiento variacional. Introduciremos un par de definiciones más, que nos permiten vincular la representación intencional de agentes con la de estados de un CBR y tendremos, de este modo, un lenguaje de representación para estructuras integradas CBR-BDI.

Definición. Dado un sistema de coordenadas canónico para un problema P , en $I(\Theta)$, denotado (A_1, A_2, \dots, A_v) , podemos reordenarlo si diferenciamos entre las variables

$$\begin{aligned} \{F_m\} &= \{A_j \text{ con } j \leq v/A_j \text{ crecientes en el proceso de solución}\} \text{ y aquellas} \\ \{G_n\} &= \{A_k \text{ con } k \leq v/A_k \text{ decrecientes en el proceso de solución}\}, \text{ de forma que} \\ &\aleph = \{F_m\} \cup \{G_n\} \text{ y } m + n = v \end{aligned}$$

horario (hr)	10-11	11-12	12-13	13-15	15-17	17-19	19-20	20-21	21-22
intención	M2	D(A, A)	R1	D(A, B)	M3	Tx(B, A)	M1	Tx(A, B)	R2
Coste(€)	6	0	12	0	12	3	12	3	12
Tiempo (hr)	1	1	2	2	2	1	1	1	1
Calidad	1	1	2	2	2	1	1	1	1

Tabla 3: Valores para la intención i_2 .

Definición. Con la reorganización en \aleph , y dada una $i \in I(\Theta)$, podemos expresar funcionalmente la doble relación de dependencia que tenemos para las intenciones:

- en términos de los atributos para creencias

$$i = i[e_1(\tau_1, \tau_2, \dots, \tau_j), e_2(\tau_1, \tau_2, \dots, \tau_k), \dots, e_s(\tau_1, \tau_2, \dots, \tau_q)] = i(\tau_1, \tau_2, \dots, \tau_n)$$

- en términos de las variables canónicas asociadas a \aleph , y reordenadas

$$i = i(A_1, A_2, \dots, A_v) = i(F_1, F_2, \dots, F_m, G_1, G_2, \dots, G_n)$$

lo cual determina una relación del tipo $A_j = A_j(\tau_1, \tau_2, \dots, \tau_n)$.

En este punto, podemos plantear la relación fundamental entre agentes BDI y sistemas CBR.

Recordemos que hemos planteado la solución $S(P)$ para un problema $P = (S_i, \{\Theta_j\}, S_f)$, como una secuencia de estados $S_k = (\{O_r\}_{r=1, \dots, p}, \{R_s\}_{s=1, \dots, q})$ interrelacionados por operadores $\{S_k, \Theta_h\}$, donde el estado inicial y final son los que determina el enunciado del problema.

Si el problema se le plantea a un agente deliberativo BDI, que habita el mundo Θ , donde definimos un sistema canónico $\aleph = (A_1, A_2, \dots, A_v)$ para el conjunto de las intenciones $I(\Theta)$ que es reordenado en $\aleph = (F_1, F_2, \dots, F_m, G_1, G_2, \dots, G_n)$, podemos establecer la siguiente analogía entre los parámetros

$$\begin{aligned} \{F_m\} &\longleftrightarrow \{O_r\} \\ \{G_n\} &\longleftrightarrow \{R_s\} \end{aligned}$$

Mediante esta relación, queda definido un criterio de identificación entre “estados intencionales”, $\rho_i \in I(\Theta)$, y “estados CBR”, $S_k \in T(BC)$.

Establecida esta relación, encontramos también un modo de hacer corresponder otros elementos actitudinales como los deseos $d \in D(\Theta)$ y el operador eficacia $\mathfrak{F}[S(P)]$ que asociamos a soluciones del CBR.

El detallado formalismo matemático que hemos ido desgranando tiene ahora explicación: el desarrollo de un lenguaje común para integrar aspectos de agentes y de estructuras CBR, en un mismo problema.

$idx(C1)$	$idx(C2)$
$\mathfrak{F}_x = 0.432$	$\mathfrak{F}_x = 0.516$
$\mathfrak{F}_y = 0.738$	$\mathfrak{F}_y = 0.7754$

Tabla 4: Parámetros de eficiencia $idx(C1), idx(C2)$.**Ejemplo:**

Si consideramos dos posibles rutas i_1, i_2 , junto con sus valores presentados en las tablas 2(a) y 3

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

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

Si nuestro sistema de coordenadas se representa por $\aleph = (A_1, A_2, A_3, A_4) = (P, T, C, Q)$ donde P (places) = lugares visitados, T (time) = tiempo empleado, C (costs) = gastos (valor de tickets), Q (quality) = calidad (valoración de la visita).

Si expresamos estas dos intenciones, en lugar de en términos de las creencias que los constituyen, en función de sus valores para los parámetros P, T, C, Q

$$i_1 \rightarrow P = 3, T = 12(\text{h}), C = 16(\text{€}), Q = 1.6$$

$$i_2 \rightarrow P = 3, T = 9(\text{h}), C = 12(\text{€}), Q = 2$$

y los valores de P, T, C, Q son representados por una estructura $S = \left(\begin{array}{l} \{O_r\}_{r=1, \dots, p} \\ \{R_s\}_{s=1, \dots, q} \end{array} \right)$ entonces cada intención puede ser considerada como un caso, con índices asociados.

$$idx(C1) = \left(\begin{array}{l} \{3, 1.6\} \\ \{6, 12\} \end{array} \right) \quad idx(C2) = \left(\begin{array}{l} \{3, 2\} \\ \{9, 12\} \end{array} \right)$$

Si un turista presenta un problema P al agente, fijando los siguientes valores:

$$S(P) = \left\{ \begin{array}{ll} P \geq 3 & Q \geq 1.5 \\ T \leq 12 \text{ h} & C \leq 10 \text{ €} \end{array} \right\}$$

Tenemos en $idx(C1), idx(C2)$ dos "soluciones CBR", ya que, según la definición, se cumple que $P \subseteq Idx(C1), Idx(C2)$

La función "deseo" que se puede expresar como

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

$$i = (e_1 \wedge \dots \wedge e_r) \rightarrow F = F(i) = \{P \geq 3, Q \geq 1.5, T \leq 12, C \leq 10\}$$

que se traducirá en restricciones al atributo $\mathfrak{F}(S(P)) = (\mathfrak{F}_x, \mathfrak{F}_y)$, y haciendo cálculos, se impone que la solución se exprese en los siguientes términos: $\mathfrak{F}_x = 0.412$, es decir, la $S(P)$ debe alcanzar al menos un 41.2 % de sus objetivos; $\mathfrak{F}_y = 0.790$, esto es, no debemos pasar del 79 % de gasto de recursos. La Tabla 4 representa los parámetros de eficiencia de los casos $idx(C1), idx(C2)$.

La relación de identificación entre "estados intencionales" y "estados CBR" presentada, nos permite plantear que agentes deliberativos, con arquitecturas BDI, puedan valerse de un ciclo de razonamiento tipo CBR, para generar soluciones $S(P)$ a proble-

mas P .

Cuando el agente necesite resolver un problema, utilizará sus creencias, deseos e intenciones para construir su solución. Deseos previos, junto con creencias e intenciones, son almacenadas en forma de casos, y se convierten en material susceptible de recuperación para nuevos problemas. Los casos que puedan ser útiles al problema actual son seleccionados, y adaptados, generando el plan de acción del agente que será la nueva solución. Esta expresión es revisada y, si cumple con las propiedades requeridas, es almacenada, lo que constituye un mecanismo de aprendizaje asociado a cada proceso.

4. Modelización de agentes CBR-BDI

En las secciones que siguen se mostrará cómo definir criterios operativos para las fases de recuperación y de adaptación en un CBR, y cómo este mecanismo permite la automatización del razonamiento. El modelo se asienta en el cálculo variacional [2, 9], y su mayor ventaja será capacitar a los agentes de un sistema de planificación y re-planificación en tiempo de ejecución.

Veamos como puede establecerse una relación operativa entre sistemas CBR y arquitecturas BDI, asociando estructuras actitudinales formadas por creencias, deseos e intenciones, con estructuras de casos.

4.1. Recuperación

Durante esta fase, un conjunto de casos, C_1, C_2, \dots, C_m , almacenados en la base de casos CB y que sean similares al problema planteado, son recuperados. Necesitamos una métrica que determine qué significa ser similar.

Definición. Dado un problema P , decimos que P' es un problema "similar" y denotamos $P' \approx P$, si el caso $C' = (P', S(P'), \mathfrak{F}[S(P')]) \in CB$ es una posible solución CBR y además cumple que $idx(C') \supseteq \{idx(C_k) | k = 1, \dots, n\}$.

Hasta aquí tenemos un conjunto de posibles soluciones CBR, $P \subseteq idx(C_1), \dots, idx(C_m)$

Usaremos el parámetro **eficiencia** $\zeta[S(P)]$, que indica la cantidad de recursos que deben ser empleados por unidad de objetivo alcanzado.

Definición. Los casos que tengan un valor de eficiencia máxima son seleccionados y se denotan por $\zeta[S(P)]_{\text{máx}}$, lo cual filtra sólo un subconjunto de todas las soluciones seleccionadas: $P \subseteq idx(C_1), idx(C_2), idx(C_3), \dots, idx(C_r)$, con $r \leq m$.

Por tanto, queda reducido el problema a identificar cuál es el mejor caso de este subconjunto.

Antes de exponer cómo determinar el caso solución, se introducen efectos de no-linealidad en la dependencia de los casos con sus atributos, como mostramos en el

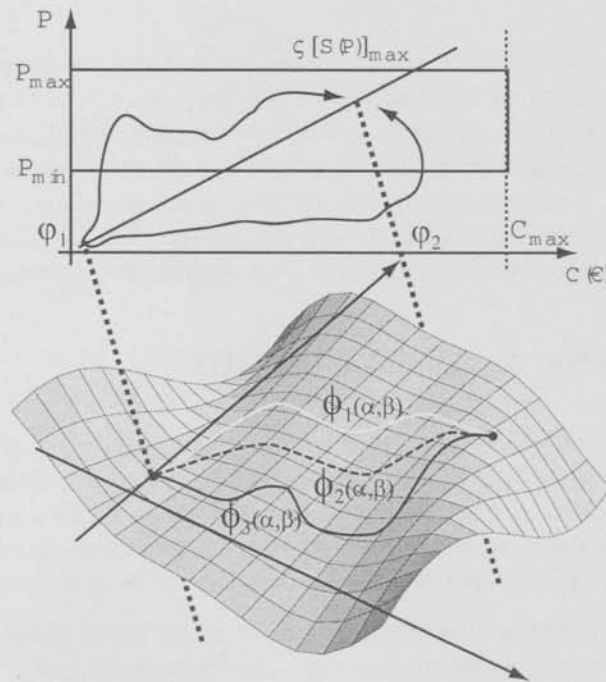


Figura 2: Efectos de no-linealidad sobre el espacio del problema.

siguiente ejemplo.

Ejemplo:

La visita a un museo $M2$, con $Q = 1$, puede costar $C = 2 \text{ €}$, mientras que la visita a $M1$ con $Q = 2$, puede ser visitada gratuitamente, $C = 0 \text{ €}$ (por ejemplo, si existe un programa público de promoción cultural).

Para incorporar la no-linealidad a nuestro problema, codificaremos todos los procesos no lineales en una función $V = V(A_1, A_2, \dots, A_v)$. El potencial definido en el espacio generado por (A_1, A_2, \dots, A_v) introduce relaciones de ligadura entre estas variables, lo que, gráficamente, puede verse como una “curvatura” sobre el estado de fases, en un espacio homogéneo, isótropo y lineal, con $V = 0$. (ver Figura 2).

Ejemplo:

En términos del agente turístico, considerando solamente $\aleph = (A_1, A_2) = (P, C)$ y un problema en el que se exija para su solución un P (mínimo) y un C (máximo). La Figura 3 representa tres potenciales soluciones ϕ_1, ϕ_2, ϕ_3 , asumiendo la no-linealidad del espacio de soluciones.

En general, para un $V = V(A_1, A_2, \dots, A_v)$ tendremos una familia de hipersuperficies que marcan las zonas del espacio “favorables al movimiento”, quedando fijada una hipersuperficie concreta al especificar dos puntos (S_i, S_f) .

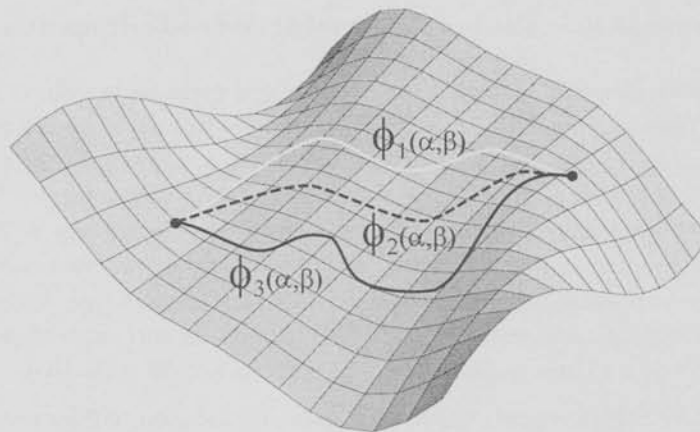


Figura 3: Representación de caminos posibles entre dos puntos.

Supondremos que las hipersuperficies cumplen las condiciones de diferenciabilidad que se le exigen a una variedad k -dimensional.

Definición. Un subconjunto M de \mathbb{R}^n es una variedad k -dimensional si y sólo si para cada punto $x \in M$ se verifica la siguiente “condición de coordenadas”: Existe un conjunto abierto U que contiene x , existe un conjunto abierto $W \subset \mathbb{R}^k$ y existe una función diferenciable $f : W \rightarrow \mathbb{R}^n$ tal que

$$f(W) = M \cap U$$

$$f'(y) \text{ tiene rango } k \text{ para cada } y \in W$$

A esta función f , con estas propiedades, se le denomina sistema de coordenadas alrededor de x . Con esto, lo que queremos es que la función potencial $V = V(A_1, A_2, \dots, A_v)$ garantice unas condiciones mínimas de diferenciabilidad a la familia de hipersuperficies que forman sus órbitas, para poder aplicar cálculo variacional.

Resumiendo, con la introducción de un potencial $V = V(A_1, A_2, \dots, A_v)$ en el espacio de trabajo, queremos separar los efectos asociados a la “curvatura”. Intuitivamente, queremos que la curvatura que introduzca sea suficientemente suave como para que podamos construir un sistema de coordenadas alrededor de cada punto que, entre otras cosas, nos permitirá tomar medidas sobre él.

Si tenemos un espacio no lineal sobre el que trazar trayectorias, y éstas son medibles, podemos plantearnos cómo resolver el problema de determinar qué trayectoria cumple condiciones de mínimo para un cierto atributo. Esa trayectoria puede representar una solución a un problema y la condición de mínimo sobre un parámetro, un criterio de utilidad típico impuesto sobre la solución. La herramienta que permite hacer esto existe y se conoce como cálculo variacional (ver Apéndice A).

4.1.1. Aplicación del cálculo variacional al modelado de agentes CBR-BDI

Propusimos en la sección anterior una notación que permitía la definición de agentes "CBR-BDI". Tales agentes presentan tanto habilidades típicas de estructuras deliberativas, planificación de acciones orientadas a un objetivo, como características propias de sistemas CBR (como el aprendizaje y la interacción con el entorno). Los sistemas CBR pueden ser implementados y automatizados de diferentes formas [6, 7] dependiendo del tipo de problemas que resuelven. En esta sección se muestra cómo el cálculo variacional es una potente herramienta en sistemas que admiten una representación en variedades –lineales y no-lineales–, y puede aplicarse en la automatización de la etapa de recuperación de un sistema CBR, integrado en una arquitectura BDI.

En general, el cálculo variacional proporciona una solución óptima pero en un conjunto de valores discretos como el de nuestro trabajo, se utilizará para obtener la solución discretizada más cercana a la óptima [14], ya que posiblemente, ésta no se alcance.

Consideremos una base de casos e índices de casos, sobre la que definimos una relación de subsumción $(CB, R) = \{[C_k/k = 1, \dots, q; \text{ donde } q \in \mathbb{N}, R]\}$.

Denotamos el conjunto de atributos de la base de casos $T(CB) = (\alpha_1, \alpha_2, \dots, \alpha_m)$, $\alpha_j \in T$. Utilizando la relación entre agentes BDI y sistemas CBR establecida en la sección anterior, denotaremos $T(CB) = (A_1, A_2, \dots, A_v)$, donde $\{A_j\}$ es un sistema de coordenadas canónico, para un problema, del conjunto $I(\Theta)$. En nuestro ejemplo, $T(CB) = (P, C)$.

Los fenómenos no-lineales definen una función V sobre este espacio $I(\Theta)$, que toma una forma $V = V(A_1, A_2, \dots, A_v)$ y que almacena información proporcionada por los mismos casos $Ck \in CB$. En el espacio de fases, la información de la función $V = V(A_1, A_2, \dots, A_v)$ se traslada a la forma espacial de una hipersuperficie $\Pi_0[A_1, A_2, \dots, A_v] = 0$, donde la noción de distancia euclídea es definida.

Si consideramos dos estados (S_i, S_f) inicial y final, sobre el conjunto $I(\Theta)$, y las intenciones $i \in I(\Theta)$, que unen estos estados y que tienen asociado un caso recuperado $Ck \in CB$, podemos definir una función distancia como

$$D(S_i, S_f) = \sqrt{(A_{i1} - A_{f1})^2 + (A_{i2} - A_{f2})^2 + \dots + (A_{im} - A_{fm})^2} \quad (4)$$

donde $S_i = (A_{i1}, A_{i2}, \dots, A_{iv})$, $S_f = (A_{f1}, A_{f2}, \dots, A_{fv})$.

En el caso $m = 3$, y denotando $A_1 = X, A_2 = Y, A_3 = Z$, la teoría variacional nos dice que existe un sistema de coordenadas (λ, μ) , sobre el cual se permite la expresión de una funcional integral $F = F(\lambda, \mu)$ que asocie a cada curva entre (S_i, S_f) su longitud, y que valore sobre $\Pi_0[x, y, z] = 0$. Además nos facilita la forma de conocer cuál de estas curvas es la óptima en el sistema de coordenadas (λ, μ) .

La resolución de la ecuación de Euler,

$$\frac{\partial F}{\partial \mu} - \frac{d}{d\lambda} \left(\frac{\partial F}{\partial \mu'} \right) = 0 \quad (5)$$

nos proporciona $\mu = \mu_0(\lambda)$ que toma la forma $\chi_0 = \chi_0[x, y, z]$ sobre el sistema de coordenadas original (X, Y, Z) . Se conoce a esta función como "curva geodésica".

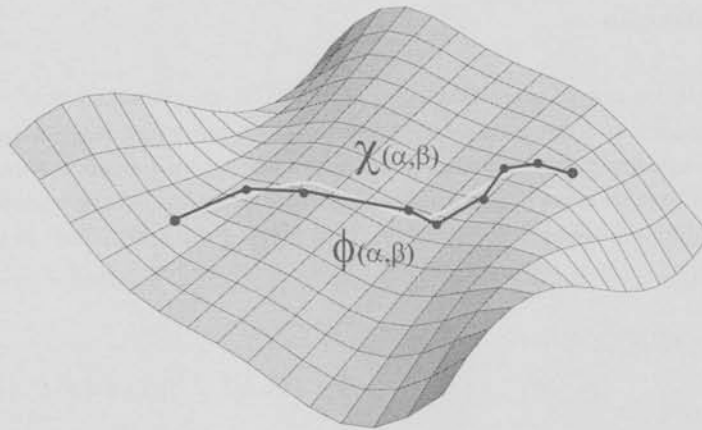


Figura 4: Curva geodésica y curva solución más cercana.

En el caso más general, la función $V = V(A_1, A_2, \dots, A_m)$ genera curvas que no son diferenciables porque V sólo toma valores correspondientes a estados discretos.

Definición. Definiremos una función σ , como $\sigma = \chi_0 - \Psi$, donde χ_0 es la solución obtenida por la ecuación de Euler (geodésica) [14] y $\Psi \in \{\varphi(S_i, S_f)\}$ es un camino entre (S_i, S_f) , asociado a uno de los casos $C \in CB$ (ver Figura 4).

Definición. Llamaremos “curva más cercana a la óptima”, y la denotamos Ψ_0 a la función de $\{\varphi(S_i, S_f)\}$ dada por la minimización de la integral

$$I = \int_{ei}^{ef} \{\sigma[X, Y, Z]\} dx dy dz \tag{6}$$

que escribimos como $\Psi_0 = \{S_i = S_0^{(0)}, S_1^{(0)}, S_2^{(0)}, S_3^{(0)}, S_4^{(0)}, \dots, S_m^{(0)}, \dots, S_s^{(0)} = S_f\}$, donde S_k son estados sobre los que estructuramos la solución. Por tanto, el cálculo variacional determina cuál es, de las “soluciones cbr” para un problema dado, aquella con la que nos quedamos. Intuitivamente, seleccionamos aquella que nos presenta la forma de llegar al estado final de manera que los estados que cruza difieran, de la menor manera posible, en sus valores de eficiencia.

Este cálculo se plantea para sistemas en condiciones estáticas, pero los agentes son sistemas dinámicos, que deben responder a cambios del medio en tiempo real, planear sus soluciones, y modificarlas si los cambios en el entorno lo requieren. En una estructura como la que presentamos, los cambios en el entorno se convierten en nuevos criterios para identificar y calcular una solución CBR.

El formalismo desarrollado se extiende fácilmente y permite definir un agente “CBR-BDI” dinámico, con capacidad de replanificación en tiempo real, si el medio y sus variaciones fuerzan a cambiar el plan en ejecución.

4.2. Reutilización

La fase anterior nos proporciona un caso de la CB, que denotamos $C' = (P', S(P'), \mathfrak{F}[S(P')])$, primero aplicando una métrica de similaridad sobre los casos de la base, y seguidamente aplicando cálculo variacional sobre el conjunto de soluciones preseleccionadas. Durante esta fase el sistema ejecuta un mecanismo de "adaptación transformacional" [1] que puede ser representado por una función de adaptación, la cual, a partir del conocimiento de la solución $S(P')$, es capaz de obtener una solución $S(P)$ para P .

$$A : (CB) \times \Sigma(P) \rightarrow C \quad (7)$$

$$(C', P) \rightarrow A(C', P) = \{P, A[S(P')], \mathfrak{F}(Ad[S(P')])\} \quad (8)$$

donde $\Sigma(P)$ se conoce como conjunto de problemas, y $C' = (P', S(P'), \mathfrak{F}([S(P')]))$.

Necesitamos explicitar qué es lo que hace la función Adaptación, con el fin de que nos sirva como criterio operativo. Algunos autores sugieren un mecanismo de adaptación en el que la función recuperada sea la más sencilla de adaptar. Es decir, establece el criterio de recuperación subordinado al de adaptación.

Nuestro enfoque es el inverso: es la fase de recuperación la que prevalece, siendo el mecanismo de adaptación puramente transformacional y reducido a una copia de la solución recuperada hasta que el contexto no permite que sea utilizada. En este momento, se regresa a la etapa anterior y se comienza un nuevo proceso de recuperación. El nuevo problema estará planteado en términos del mismo estado final, pero con un nuevo estado inicial, aquel en el que la solución previa se ha detenido.

Veámoslo en detalle, si $P = (S_i, S_f)$ y la fase de recuperación nos dice que existe un caso $C = \{P', S(P'), \mathfrak{F}[S(P')]\} \in (CB)$ candidato a ser solución, la función de adaptación construye una solución para P que es la misma secuencia de operadores que en $S(P')$ mientras existan creencias que soporten esta solución. En el momento en que esto no pueda seguir adelante, se genera un nuevo ciclo de recuperación y, con la solución recuperada, se inicia un nuevo proceso de adaptación; así sucesivamente hasta que alcancemos el estado final.

Por tanto, la función adaptación A total, puede entenderse como una serie de operadores $A = \alpha_m \bullet \alpha_{m-1} \bullet \dots \bullet \alpha_2 \bullet \alpha_1$, cada uno de los cuales es parte de un caso recuperado.

Ejemplo:

Tenemos un problema $P = (S_i, S_f)$, y existe un $P' = (S'_i, S'_f)$ con $S(P') = (S_1 = S'_i, \theta'_1, S'_2, \theta'_2, \dots, \theta'_n, S'_{n+1} = S'_f)$ tal que puede ser utilizado como solución a adaptar.

La función $A[S(P')] = \{S_i = S_1, \theta''_1, S''_2, \theta''_2, S''_3, \theta''_3, \dots, \theta''_j, S''_{j+1}\}$ comienza en el estado inicial de P y utiliza los operadores θ_j de $S(P')$ hasta que el contexto le permite.

Tenemos una solución explícita para un problema intermedio, que no llega a ser P , y que denotaremos $P1 = (S_i, S''_{j+1})$. Alcanza un estado S''_{j+1} que será el punto de arranque para un problema denotado $P'' = [S''_{j+1}, S_f]$ sobre el que comenzaremos de nuevo.

Denotamos

$$\alpha 1[S(P')] = \{S_i = S_1, \theta'_1, S''_2, \theta'_2, S''_3, \theta'_3, \dots, \theta'_j, S''_{j+1}\} = S(P1)$$

supongamos que para el nuevo problema planteado, $P'' = [S''_{j+1}, S_j]$, encontramos un nuevo caso que permita alcanzar un estado S tal que $S''_{j+1} \leq S \leq S_j$. Llamando $P2 = [S''_{j+1}, S]$, tendríamos un operador $\alpha 2[S(P'')] = S(P2)$ y así sucesivamente hasta alcanzar el estado S_F , podemos decir que,

$$\begin{aligned} A[S(P')] &= (\alpha m \bullet \alpha m - 1 \bullet \dots \bullet \alpha 2 \bullet \alpha 1)[S(P')] = (\alpha m \bullet \alpha m - 1 \bullet \dots \bullet \alpha 2)[S(P'')] \\ &= \dots = (\alpha m)[S(P^m)] = S(P) \end{aligned}$$

4.3. Revisión

En esta fase, la solución generada en la etapa de adaptación es evaluada y revisada. Tenemos, para un problema P , una solución $S(P)$ con $\mathfrak{F}[S(P)]$. En esta fase se comprueba que el proceso ha sido ejecutado adecuadamente y que se garantiza que

$$\mathfrak{F}\{A[S(P')]\} \supseteq \mathfrak{F}[S(P)] \tag{9}$$

si la relación se cumple, el proceso de revisión está completado.

4.4. Memorización

Si la fase de revisión se resuelve positivamente, las características del problema y de la solución obtenida son memorizadas como un nuevo caso, que formará parte del material almacenado para ser usado en el futuro, y se denota

$$C = \{P, A[S(P')], \mathfrak{F}(A[S(P')])\} = (P, S(P), \mathfrak{F}[S(P)]) \tag{10}$$

5. Planificación y replanificación dinámica

Hasta este punto, hemos definido un criterio que nos permite seleccionar una solución entre las soluciones de problemas almacenados, para posteriormente ser adaptada a un nuevo problema. El criterio presentado, basado en cálculo variacional, adquiere ahora su verdadera dimensión, pues es sencillo extenderlo para posibilitar un modo de planificación estratégico, es decir, un mecanismo que decide ahora, pero con vistas a posibles cambios en el futuro.

Asumiendo que los cambios significativos pueden ser determinados después de ejecutar una acción primitiva, entonces es posible controlar el dinamismo de nuevos eventos en el dominio y, de este modo, plantear una apropiada reconsideración del problema [8]. En la última parte de esta sección, se citará un formalismo más compacto para casos dinámicos, basado en funciones de Green, que puede ser usado para incorporar modificaciones al plan en ejecución, mediante conocimiento actualizado acerca de la evolución de los parámetros sobre los que se describe el problema. Si es aceptado que existen

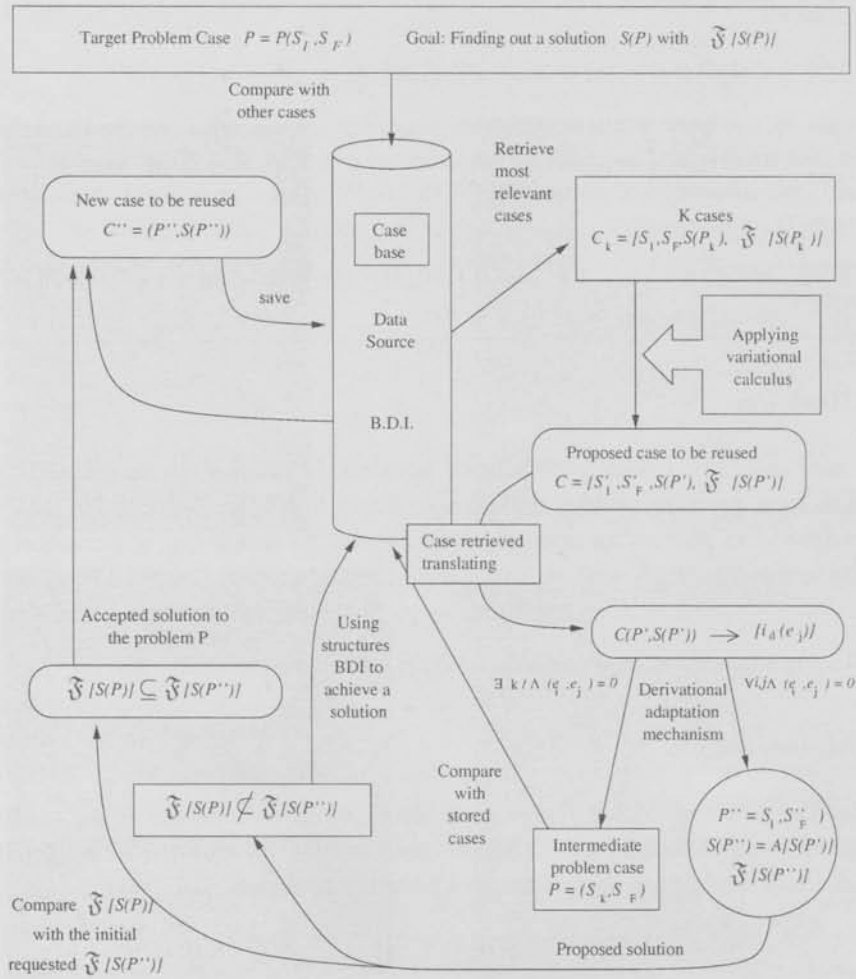


Figura 5: Esquema detallado del modelo formal.

cambios en el entorno es necesario definir mecanismos de razonamiento capaces de tratar con tales cambios y ayudar a planificar de nuevo las intenciones. Formalmente, el proceso de razonamiento puede mantenerse en términos similares, ya que la descripción que genera el problema permanece constante, como explicamos a continuación.

Si en t_0 la función $V(X, Y, Z)$ toma la forma $V_0(X, Y, Z)$, para t_1 , V se denota como $V_1(X, Y, Z)$, y denotamos la familia de hipersuperficies generadas en el espacio de fases como $\Pi_1(X, Y, Z) = 0$; sobre ésta es posible obtener la curva óptima, previamente fijados dos puntos (S_i, S_f) donde:

$$\begin{cases} S_i = S_i^{(0)}, \text{ siendo } S_i^{(0)} \text{ el estado número dos de la solución } \Psi_0 = \{S = S_0^{(0)}, \dots, S_s^{(0)} = S_f\} \\ S_f \text{ es el estado final o estado solución del problema inicial.} \end{cases}$$

Resolviendo la ecuación de Euler, se obtiene la expresión $\chi_1 = \chi_1(X, Y, Z)$ la cual

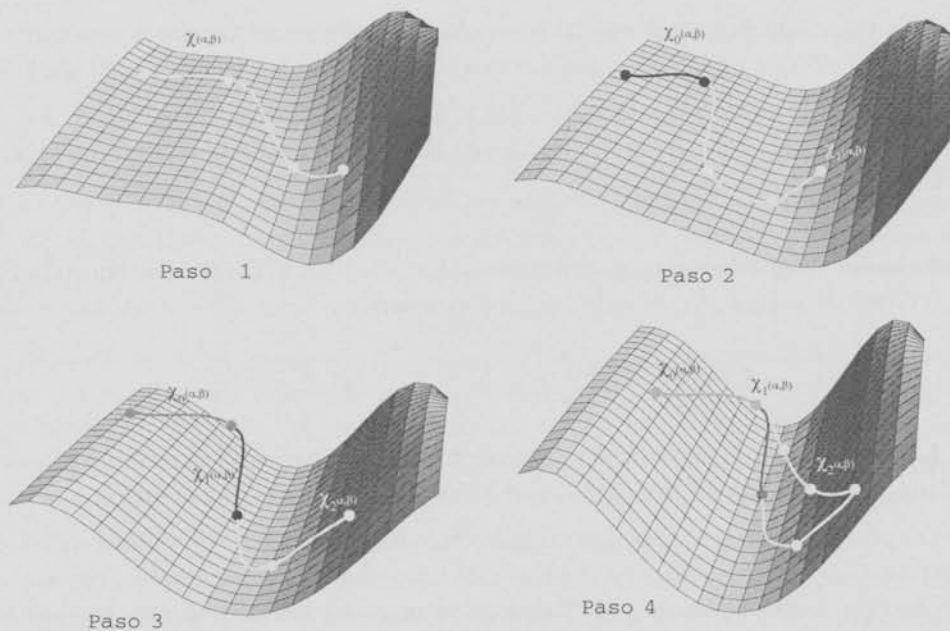


Figura 6: Representación tri-dimensional de la dinámica del entorno.

se utiliza para calcular la forma de Ψ_1 , a través de la función

$$\Psi_1 = \{S_i = S_1^{(0)}, S_1^{(1)}, S_2^{(1)}, S_3^{(1)}, S_4^{(1)}, \dots, S_m^{(1)}, \dots, S_s^{(1)} = S_f\} \quad (11)$$

Y así sucesivamente puede seguir haciéndose para cualquier t_j (ver Figura 6).

Definición. De las ecuaciones anteriores, y basándose en herramientas de cálculo variacional, estamos en condiciones de ofrecer una expresión de la intención final que represente el plan de un agente CBR-BDI, que tiene la siguiente forma:

$$\Psi_{final} = \begin{cases} \Psi_0 & \dots & t \in (t_0, t_1) \\ \vdots & \vdots & \vdots \\ \Psi_{s-1} & \dots & t \in (t_0, t_1) \\ \Psi_s & \dots & t \in (t_0, t_1) \end{cases} \quad (12)$$

La teoría de funciones de Green pueden ser usadas para la automatización de este proceso de planificación dinámica. Si la evolución del sistema puede determinarse a priori, entonces tenemos las condiciones adecuadas para aplicar un operador de Green sobre un potencial dependiente del tiempo, proporcionando una forma funcional para la solución global [14].

Con las funciones obtenidas previamente: $V_0(X, Y, Z), V_1(X, Y, Z), V_2(X, Y, Z), \dots, V_r(X, Y, Z)$ para $t_0, t_1, t_2, \dots, t_r$ podemos definir la función siguiente.

Definición. Llamaremos potencial generador dependiente del tiempo, y denotaremos $V = V[X, Y, Z, t]$, a una función que asocia a cada instante de tiempo t_j un $V_j[X, Y, Z]$.

Por ejemplo, para t_0 , la función $V[X, Y, Z, t_0]$ toma la forma $V_0[X, Y, Z]$, lo que proporciona la solución $\chi_0 = \chi_0[X, Y, Z] = [X, Y, Z, t_0]$, después de resolver la ecuación de Euler correspondiente.

Definición. Definimos el propagador de una función $V[X, Y, Z, t]$ y denotamos $G(X, X', t, t_0)$ donde $X = [x, y, z]$, $X' = [x', y', z']$ a la ecuación,

$$G(X, X', t, t_0) = \int \chi[X', t_0] V(X, t) \chi[X', t_0] dx' \quad (13)$$

La propagación de una función desde el instante de tiempo t_0 al instante t es controlada por la función $G(X, X', t, t_0)$, que hemos llamado propagador.

Los propagadores son conocidos también como funciones de Green, y almacenan toda la información concerniente a la dinámica de un problema. De este modo, concluimos esta sección mencionando la posibilidad de plantear un formalismo más compacto a partir de la teoría de funciones de Green, y que pueda ser usado para automatizar y definir el mejor plan para un agente que desea realizar sus deseos.

$$\chi[X, t] = \int G(X, X', t, t_0) \chi[X', t_0] dx' \quad (14)$$

6. Caso de estudio: Agentes CBR-BDI para el turismo electrónico

El desarrollo teórico que hemos expuesto se ha aplicado en la definición, diseño e implementación de un sistema basado en agentes, que presenta comportamiento flexible, abierto, y con acceso móvil, para la promoción y gestión del turismo cultural y de interior. El sistema será estandarizado e interlingua. El diseño se dirige tanto a comercio B2B como B2C, a soporte para usuarios privados, empresas y organismo públicos, conectados directa o indirectamente con el turismo, buscando alcanzar mejoras de calidad en lo servicios.

La plataforma desarrollada ha sido específicamente diseñada para la promoción del turismo cultural, actividades de recreo y nuevas perspectivas sobre fuentes de interés patrimonial. Es combinada con información de tipo horizontal y vertical, sobre acomodación en hoteles, restaurantes, datos sobre el sector comercial y transporte, con el fin de poder atender a las necesidades de potenciales turistas, responder a sus consultas y dar soluciones a sus demandas de información y reserva en el momento en que son expresadas.

El proyecto pretende desarrollar soluciones innovadoras, prácticas y multidisciplinarias, con las que ayudar a usuarios desde cualquier localización a organizar los diferentes servicios –ofrecidos frecuentemente de manera caótica y desde diferentes fuentes– dentro de un único, dinámico e interconectado sistema de conocimiento. Con esta finalidad,

integramos nuestro sistema en una plataforma TIC que facilite el ciclo de gestión de conocimiento en varias organizaciones.

Una de las etapas iniciales en este ambicioso proyecto es el desarrollo de una arquitectura de agentes que sea útil para modelar agentes autónomos.

En esta línea, nuestro primer experimento ha sido diseñar un agente individual, usando la formalización presentada en las secciones previas, con la misión de asistir a turistas en la identificación de un horario óptimo para visitar Salamanca. Los usuarios del sistema interactúan con los agentes vía Internet o dispositivos móviles.

El mecanismo que se sigue es el siguiente: los turistas seleccionan un número de ítems, que corresponden a monumentos o lugares representativos para visitar, pudiendo fijar alguno en particular de manera explícita, seleccionan restaurantes... y un coste máximo de la visita, que esté dispuesto a asumir. Con estos datos el agente propone un horario óptimo, pero lo rectificará si es necesario, con sólo recibir nueva información facilitada por el usuario.

Las creencias, deseos e intenciones del agente son almacenadas en forma de casos –en nuestro problema los casos reflejan experiencias pasadas de anteriores turistas–. El objetivo del agente será alcanzar un plan acorde con los requerimientos de eficacia y eficiencia facilitados por el usuario.

Los turistas deciden qué valores para los atributos son aceptables y cuando, por tanto, una solución es adecuada. A través de los parámetros habilitados, deciden cuántos recursos están dispuestos a gastar, y cuantos objetivos consideran aceptable lograr. Esto implica una medida del gasto por objetivo que están dispuestos a soportar. El agente recupera los casos que pueden usarse para generar un plan, y el cálculo variacional sirve como controlador, tanto de la fase de recuperación, como del núcleo de la replanificación dinámica.

7. Conclusiones

La arquitectura CBR-BDI resuelve uno de los problemas clásicos de arquitecturas deliberativas BDI como es la falta de capacidad de aprendizaje. El ciclo de razonamiento de un sistema CBR ayuda a los agentes a resolver problemas, facilita su adaptación a los cambios en el entorno e identifica nuevas posibles soluciones.

Nuevos casos son continuamente introducidos y los más atrasados se van eliminando. El componente CBR de una arquitectura, por tanto, proporciona un camino para la manipulación y actualización del conocimiento y de las experiencias previas.

La propuesta presentada en este trabajo reduce el *gap* que existe entre la formalización y la implementación de agentes BDI. Lo que nosotros proponemos es definir creencias, deseos e intenciones –sin necesidad de un modelo simbólico ni completamente lógico– y utilizar esta representación en el ciclo de vida de un motor de razonamiento CBR que facilite la implementación directa del agente.

Ha sido introducido un formalismo riguroso que permite la integración de los dos

modelos, el de agencia y el de razonamiento basado en casos y que encuentra en el soporte matemático basado en el cálculo variacional un mecanismo de automatización del ciclo CBR, ya presentado.

Los agentes necesitan responder en tiempo real a las peticiones de los usuarios y adaptar las soluciones en tiempo real. Obtenemos del cálculo variacional el plan más adecuado para alcanzar un objetivo en entornos con incertidumbre, y se corresponde con aquel que mantiene la eficiencia entre sus estados lo más constante posible.

Hemos mostrado también cómo la arquitectura propuesta puede ser usada en el desarrollo de un agente para el turismo electrónico. El diseño presentado es la primera fase de un proyecto más ambicioso que se materializará en una comunidad de agentes capaces de resolver problemas de una manera autónoma e inteligente. Aunque la arquitectura y la formalización descrita ha sido aplicada al turismo electrónico, creemos que podría ser usada en cualquier otro dominio en el que agentes con capacidades de aprendizaje y de adaptación sean requeridos.

Nuestra propuesta ha tratado de integrar los avances tecnológicos actuales en el mundo de la informática distribuida, la inteligencia artificial, la tecnología Wireless e Internet. La multidisciplinaridad, en cuanto a los componentes de esta propuesta, es su elemento más innovador y distintivo y a nuestro modo de ver el mayor exponente de su éxito. El prototipo construido ha demostrado ser una herramienta muy útil para la organización de visitas turísticas en Salamanca, ciudad caracterizada por acoger visitantes que practican turismo cultural y que en un alto porcentaje acuden a la ciudad con dispositivos móviles desde los que pueden planificar sus visitas.

A. Introducción al cálculo variacional

Supongamos un espacio m -dimensional $X = (X_1, X_2, \dots, X_m)$ y una función sobre X , que llamamos $V(X)$, y que se denota genéricamente

$$V : X \rightarrow X \quad (15)$$

$$(X_1, X_2, \dots, X_m) \rightarrow V(X_1, X_2, \dots, X_m) \quad (16)$$

sobre el espacio de fases –llamamos así al conjunto de todos los estados de proceso– la función $V(X_1, X_2, \dots, X_m)$ se traduce en una familia de hipersuperficies $m-1$ dimensionales que denotaremos como $G(X_1, X_2, \dots, X_m) = 0$.

Consideremos dos puntos (e_i, e_f) que valoran sobre una hipersuperficie $G(X_1, X_2, \dots, X_m) = 0$, esto es, $e_i = (X_{i1}, X_{i2}, \dots, X_{im})$ cumple $G(e_i) = 0$, $e_f = (X_{f1}, X_{f2}, \dots, X_{fm})$ cumple $G(e_f) = 0$.

Definición. Definimos el conjunto $\varphi(e_i, e_f)$ que está formado por todas las posibles curvas que unen (e_i, e_f) y que denotamos $\{\varphi_1(e_i, e_f), \varphi_2(e_i, e_f), \dots, \varphi_m(e_i, e_f)\}$ y que cumplen la condición de estar “sobre la hipersuperficie”. Esto es, $\forall \varphi_j(X_1, X_2, \dots, X_m) \in \{\varphi(e_i, e_f)\}$ se satisface que $G[\varphi_j(X_1, X_2, \dots, X_m)] = 0$.

Veremos a continuación como, dado este conjunto $\varphi(e_i, e_f)$, el cálculo variacional, nos permite determinar la curva óptima respecto a su longitud, -técnicamente "geodésica"- y puede ser calculada automáticamente [2, 9, 14].

Para exponer la teoría, nos restringiremos a $m = 3$ (Figura 3), y denotaremos $X = (X1, X2, X3) = (X, Y, Z)$.

Definición. Una funcional $A[y = y(x)]$ definida sobre un espacio F , es una función continua de F en los números reales,

$$A : \Omega^\infty(\mathbb{R}) \longrightarrow \mathbb{R} \tag{17}$$

$$y = y(x) \longrightarrow A[y = y(x)] \tag{18}$$

donde $\Omega^\infty(\mathbb{R})$ es el conjunto de funciones infinitamente diferenciables en \mathbb{R} .

Dada una funcional cualquiera A , se puede demostrar que las soluciones extremales (máxima o mínima) para A , son funciones $y = y_0(x)$ tales que su variación es nula, $\delta A[y(x)] = 0$.

Definición. Llamaremos funcionales integrables si admiten una expresión con la forma

$$A[y(x)] = \int_{x1}^{x2} [F(x, y(x), y'(x))] dx \tag{19}$$

Para estos casos, la condición de extremalidad $\delta A[y(x)] = 0$ se transforma en un criterio más práctico; se dice que una función $y = y_0(x)$ es óptima para la funcional integrable $A[y(x)]$ si la función $F[x, y(x), y'(x)]$ satisface la ecuación de Euler [2]:

$$A[y(x)] \text{ extremal} \iff \delta A[y(x)] = 0 \iff \frac{\partial F}{\partial y} - \frac{d}{dx} \left(\frac{\partial F}{\partial y'} \right) = 0 \iff y = y_0(x) \tag{20}$$

Definición. Definimos sobre una superficie $G(X, Y, Z) = 0$ generada a partir de la función potencial $V = V(X, Y, Z)$, la noción de distancia euclídea que asocia a cada par de puntos (e_i, e_f) un número real $D(e_i, e_f)$ que se obtiene como,

$$D(e_i, e_f) = \sqrt{(X_i - X_f)^2 + (Y_i - Y_f)^2 + (Z_i - Z_f)^2} \tag{21}$$

donde $e_i = (X_i, Y_i, Z_i), e_f = (X_f, Y_f, Z_f)$. De esta forma, la longitud de una curva cualquiera viene dada por la siguiente expresión,

$$L = \int_{x1}^{x2} [dl] = \int_{x1}^{x2} \sqrt{dx^2 + dy^2 + dz^2} = \int_{x1}^{x2} \sqrt{1 + y'(x)^2 + z'(x)^2} dx \tag{22}$$

La relación que $G(X, Y, Z) = 0$ establece entre sus variables, implica constricciones entre $y(x), z(x)$. Esto permite la definición de dos nuevas coordenadas (ρ, Θ) sobre las que la ecuación anterior adopta una forma más sencilla para ser resuelta,

$$L = \int_{\Theta_1}^{\Theta_2} \sqrt{1 + \rho(\Theta)^2} d\Theta = L\{\Theta, \rho(\Theta), \rho'(\Theta)\} \tag{23}$$

Ya que esta expresión presenta una dependencia funcional sobre la que podemos aplicar la ecuación de Euler –es una funcional integrable–. Hemos expuesto el caso particular en 3 dimensiones, pero la generalización de la ecuación de Euler existe y es válida para cualquier número de parámetros. Para un caso general, la solución se obtiene tras resolver un sistema de n ecuaciones diferenciales. Con este ejemplo pretendemos introducir una de las claves de nuestro planteamiento.

Establecemos una analogía estrictamente formal entre las trayectorias para el movimiento en espacios “deformados” por potenciales y las rutas intencionales que se generan como planes, sobre un espacio de casos ya experimentados, para cubrir los deseos de un agente BDI. De esta forma, tenemos un aparato matemático robusto que puede aplicarse en la definición de un criterio de recuperación para un ciclo CBR. El algoritmo de recuperación se basa en las herramientas presentadas de cálculo variacional.

Referencias

- [1] A. Aamodt y E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AICOM*, 7(1), Marzo, 1997.
- [2] V.I. Arnold. *Ordinary Differential Equations*. Springer-Verlag, 1971.
- [3] M.E. Bratman. *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A., 1987.
- [4] D. Camacho, D. Borrajo, y J.M. Molina. Intelligence travel planning: a multiagent planing system to solve web problems in the e-tourism domain. *International Journal on Autonomous agents and Multiagent systems*, 4(4):385–390, Diciembre, 2001.
- [5] P.R. Cohen y H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3), 1990.
- [6] J.M. Corchado y B. Lees. A hybrid case-based model for forecasting. *Applied Artificial Intelligence*, 15(2):105–127, 2001.
- [7] C. Fyfe y J.M. Corchado. Automating the construction of cbr systems using kernel methods. *International Journal of Intelligent Systems*, 16(4), 2001. ISSN 0884-8173.
- [8] M. González y J.M. Corchado. A planning strategy based on variational calculus for deliberative agents. *Computing and Information Systems Journal*, 10(1), 2002. ISBN: 1352-9404.
- [9] M.H. De Groot. *Optical Statistical Decisions*. McGraw-Hill, New York, 1970.
- [10] D. Kinny y M. Georgeff. Commitment and effectiveness of situated agents. En *In Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI91)*, pp 82–88, Sydney, Australia, 1991.

- [11] C.A. Knoblock, S. Minton, J.L. Ambite, M. Muslea, J. Oh, y M. Frank. Mixed-initiative, multisource information assistants. En *10th International world wide web conference (WWW10)*. ACM Press, May 1-5, 2001.
- [12] R. Laza y J.M. Corchado. Construction of bdi agents from cbr systems. En *First German Workshop on Experience Management*, Vol. 1, pp 47-60, Berlin, Marzo, 2002. ISBN 3-88579-340-7.
- [13] F.J. Martín, E. Plaza, y J.L. Arcos. Knowledge and experience reuse through communications among competent (peer) agents. *International Journal of Software Engineering and Knowledge Engineering*, 9(3):319-341, 1999.
- [14] P.M. Morse y H. Feshbach. *Methods of Theoretical Physics*. McGraw-Hill, New York, 1953.
- [15] C. Olivia, C.F. Chang, C.F. Enguix, y A.K. Ghose. Case-based bdi agents: An effective approach for intelligent search on the world wide web. En *AAAI Spring Symposium on Intelligent Agents*, Stanford University, Marzo 22-24, 1999.
- [16] A.S. Rao y M.P. Georgeff. Bdi agents: From theory to practice. En *First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USA, June, 1995.
- [17] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51-92, 1993.
- [18] I. Watson y F. Marir. Case-based reasoning: A review. *The knowledge Engineering Review*, 9(3), 1994. Cambridge University Press.
- [19] J. Wendler y M. Lenz. Cbr for dynamic situation assessment in an agent-oriented setting. En *Proc. AAAI-98 Workshop on CBR Integrations*, Madison (USA), 1998.
- [20] M. Wooldridge y N.R. Jennings. Agent theories, architectures, and languages: A survey. En *Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, 1994.