

Modelado de características para itinerarios formativos adaptativos.

J. M. Marquez¹, C. Cetina², F. Velasco³, L. Gonzalez-Abril³, J. A. Ortega⁴

¹Dep. de I+D+i. Telvent, R. Tamarguillo 29, Sevilla. jose.marquez@telvent.abengoa.com

²Dep. de Sistemas Informáticos y Computación, Univ. P. de Valencia, ccetina@dsic.upv.es

³Dep. de Economía Aplicada, Univ. de Sevilla. {velasco,luisgon}@us.es

⁴Dep. de Lenguajes y Sistemas Informáticos, Univ. de Sevilla, ortega@lsi.us.es

Abstract

Este trabajo presenta una forma de combinar dos técnicas diferentes de la inteligencia artificial para aplicarlas en la adaptación de itinerarios formativos usando plataformas de e-Learning. La utilización de las técnicas descritas en este trabajo, a diferencia de otras técnicas más "intrusivas", no afecta al formato de empaquetado de los contenidos y mantiene la compatibilidad de éstos con las especificaciones existentes en la actualidad, permitiendo la adaptación del currículo en tiempo de ejecución, y adaptándolo a las necesidades específicas de los alumnos.

1 Introducción

Hoy en día, las especificaciones existentes no tienen en cuenta cómo gestionar diferentes cursos independientes para formar itinerarios formativos complejos y las posibilidades de adaptar el itinerario formativo a las necesidades personales de cada alumno son mínimas. De esta forma, IMS-CP, la especificación más ampliamente aceptada por la comunidad de e-Learning para el empaquetado de contenidos y piedra angular de especificaciones más complejas como SCORM o IMS-LD, se centra en la descripción de los recursos educativos que se incluyen en un paquete o archivo comprimido en formato ZIP que puede distribuirse a multitud de plataformas de gestión del aprendizaje (LMS) compatibles, pero no tiene en cuenta las posibles relaciones que pudieran establecerse entre diferentes paquetes para dar lugar a estrategias de aprendizaje adaptadas a las necesidades del alumno. Estas especificaciones se centran en los procesos de aprendizaje involucrados en un único paquete de contenidos, pero no en situaciones en las que intervienen más de uno. Aunque las especificaciones actuales incluyen en su mayoría especificaciones para gestionar el secuenciamiento de las unidades didácticas incluidas en un paquete, el desarrollo de diferentes versiones de cada una de estas unidades didácticas, atendiendo a diferentes niveles de detalle y complejidad, y su distribución en diferentes paquetes de contenidos abre un abanico de posibilidades para adaptar el proceso de aprendizaje al ritmo del alumno.

Algunos autores [Rey-López et al., 2006] han centrado sus trabajos en la adaptabilidad de contenidos, proporcionando buenas aproximaciones para incorporar reglas de adaptación en los descriptores de paquetes SCORM. El secuenciamiento de estos paquetes a lo largo de procesos de formación más amplios es llevado a cabo en lado del servidor, generalmente por aplicaciones web encargadas de aplicar los mecanismos de adaptabilidad adecuados. Esta es una solución relativamente sencilla de modelar cuando el camino de aprendizaje, compuesto por la lista de cursos, formada por una lista de paquetes de contenidos educativos, es estática y ha sido establecida previamente, y las adaptaciones posibles se reducen a la reproducción de una parte u otra de cada curso, en función de las necesidades y/o preferencias del alumno. Sin embargo, se requiere un gran esfuerzo computacional cuando la lista de cursos que forman el itinerario formativo no es estática y puede variar durante la realización del mismo. Esto puede suceder cuando son los propios paquetes o cursos los que pueden variar según las necesidades y/o preferencias de los alumnos, independientemente de que a cada curso seleccionado puedan serle de aplicación mecanismos de adaptabilidad para reproducir parcialmente unas u otras de sus unidades didácticas.

En nuestro enfoque, el itinerario formativo puede ser personalizado dependiendo de las capacidades propias de cada alumno, así como del comportamiento o capacidades medias del resto de alumnos que comenzaron el mismo itinerario. Este enfoque muy relacionado con la variabilidad en los sistemas software, el aprendizaje automático, la toma de decisiones y aplicado a un entorno educativo, permitirá una mejor atención a la diversidad del alumnado.

2 Escenario

Supongamos que una empresa quiere preparar a sus empleados para adaptarlos a un puesto diferente que queda vacante de mayor responsabilidad. Esta vacante requiere una serie de competencias que pueden ser adquiridas por los candidatos mediante la realización de algunos cursos específicos.

Nuestra aproximación consiste en una plataforma educativa modular y escalable, cuyos componentes, incluidos los cursos se distribuyen como bundles OSGi. Tal y como se detalla en un trabajo previo [Márquez, 2007], las especifica-

ciones que definen el empaquetado de contenidos educativos, SCORM, y el empaquetado de aplicaciones y servicios, OSGi, no son incompatibles. Partiendo de esta compatibilidad y partiendo de otros trabajos centrados en la definición y resolución de dependencias entre componentes OSGi [Ruiz et al., 2004] y el modelado de la variabilidad en sistemas software sería posible construir un árbol que describa todos los caminos posibles del itinerario formativo necesario para adquirir las competencias requeridas para la vacante, teniendo en cuenta las necesidades y preferencias de cada candidato individualmente, así como las del conjunto de ellos, para personalizar el proceso de aprendizaje de cada uno ellos al ritmo que mejor se adecue a las características de cada uno, maximizando la probabilidad de éxito y el nivel de adquisición de los conocimientos y habilidades requeridas para el puesto vacante, objetivo esencial de la formación.

El proceso implica la superación satisfactoria de un conjunto de cursos, cada uno distribuido como un bundle OSGi. Sin embargo, este bundle no es único: el bundle y la complejidad de los contenidos asociados a un curso puede variar de un usuario a otro, dependiendo del perfil de aprendizaje del candidato. La empresa desea que los cursos puedan variar dinámicamente reaccionando al proceso de aprendizaje de cada usuario, variando la duración total del itinerario para adaptarla al ritmo de aprendizaje de cada uno, modificando la proporción de contenidos prácticos y teóricos, introduciendo cursos de refuerzo de contenidos vistos en otros anteriores, etc. Este comportamiento del sistema ayudará a los usuarios a adquirir las capacidades necesarias al ritmo que más se adapte a su estilo de aprendizaje, tiempo disponible, etc.

Esta plataforma inteligente podría recomendar o incluso imponer el curso siguiente, teniendo en cuenta los resultados de cada candidato, así como el itinerario más exitoso seguido por la mayoría de ellos, por lo que gana en interés cuantos más usuarios realicen un mismo itinerario simultáneamente.

2.1 Descripción de Itinerarios de Aprendizaje

Con este objetivo, el equipo pedagógico de la compañía tiene que diseñar todas las posibles variantes de caminos que conformaran el itinerario formativo. Para dar soporte a esta tarea proponemos aplicar el enfoque de las Líneas de Producto Software (LPS) [Batory, 2006]. Las LPS definen una familia de productos similares a partir de la cual es posible derivar productos distintos entre sí. En nuestro caso la familia de productos se corresponde con el itinerario formativo, mientras que cada posible producto será un camino distinto. En esta sección abordamos como utilizar la técnica de Modelado de Características de las LPS para describir un itinerario de aprendizaje.

La técnica de Modelado de Características [Czarnecki et al., 2004] se utiliza para describir un sistema y sus variantes en términos de características. Estas características se estructuran mediante relaciones de variabilidad y adicionalmente es posible definir restricciones entre ellas. Aplicando esta técnica al dominio de los itinerarios formativos es posi-

ble modelar todos los posibles caminos existentes en un itinerario. Las características representan los cursos académicos, mientras las relaciones entre características representan las posibles transiciones validas entre cursos. En concreto, estas transiciones introducen los siguientes conceptos:

- **Opcional.** Un curso es opcional cuando una vez superados los cursos previos este puede ser cursado o no cursado por el alumno. Gráficamente se representa con un círculo blanco sobre el curso.
- **Obligatorio.** Un curso es obligatorio cuando una vez superados sus cursos previos el alumno obligatoriamente debe cursarlo. Gráficamente se representa con un círculo negro sobre el curso.
- **Relación AND.** Un conjunto de cursos tiene una relación AND con un curso previo cuando una vez superado este todos los cursos del conjunto han de ser cursados por el alumno. Gráficamente se representa con un triángulo negro.
- **Relación OR.** Un conjunto de cursos tiene una relación OR con un curso previo cuando una vez superado este el alumno debe cursar únicamente un curso del conjunto. Gráficamente se representa con un triángulo blanco.

Adicionalmente, entre dos cursos es posible definir relaciones que establecen restricciones sobre el camino de aprendizaje.

- **Requiere.** Una relación de la forma A requiere B establece que en todo camino de aprendizaje donde A este presente necesariamente B también lo estará. Gráficamente se representa con una flecha discontinua.
- **Excluye.** Una relación de la forma A excluye B establece que en todo camino de aprendizaje donde A este presente necesariamente B no lo estará. Gráficamente se representa con una flecha discontinua de dos puntas.

En un itinerario formativos, la secuenciación de los cursos es determinada por (1) las relaciones de transacción y (2) un criterio de ordenación. El criterio de ordenación establece que el modelo de características se recorre primero en profundidad y de izquierda a derecha. Para calcular los posibles caminos, siempre se parte de una característica denominada *root* en el itinerario. La característica *root* se diferencia de las otras características en que es la única que no es destino de ninguna transición, gráficamente suele aparecer en la parte superior del modelo.

La Figura 1 muestra un itinerario formativo representado con un modelo de características. Para obtener un posible camino sobre la característica *root* ha de aplicarse el criterio de ordenación respetando las relaciones de transición. En este modelo la característica *root* se corresponde con el número 1 y un posible camino de aprendizaje valido es la secuencia {1, 2, 3, 5, 4, 10, 11, 12}. Distintas decisiones en las

relaciones de transición definirán caminos alternativos. Por ejemplo, la secuencia anterior puede ser modificada seleccionando el curso 6 en la relación alternativa que existe tras el curso 3, obteniendo la secuencia {1, 2, 3, 6, 9, 8, 4, 10, 11, 12}.

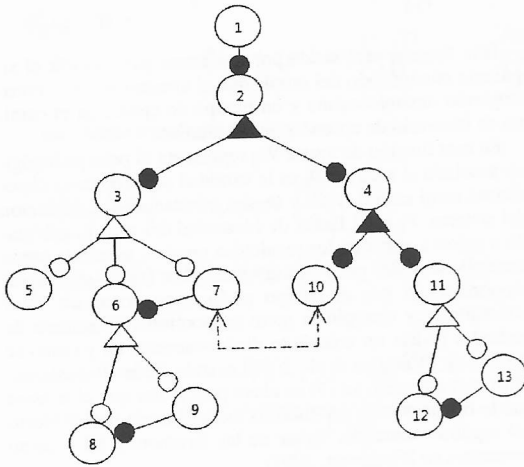


Figura 1. Modelo de Características del Itinerario Formativos.

Como herramienta de soporte para describir los itinerarios formativos hemos utilizado MOSKitt Feature Model¹ por proporcionar expresividad suficiente para describir los itinerarios y disponer de un editor gráfico. Además, dada la naturaleza open source de MOSKitt planeamos introducir extensiones a este editor que ayuden al equipo pedagógico en las tareas de definición y selección de caminos.

3 Descripción Formal utilizando XML

El Modelo de Características que describe un itinerario formativo es almacenado siguiendo el estándar XML Metadata Interchange del Object Management Group para intercambio de metadatos.

Como se muestra en la Figura 2, para cada característica del modelo se describen sus atributos y relaciones en las que participa como origen. Los atributos (etiquetados con *Attributes*) se representan mediante tuplas nombre-valor proporcionando información sobre los cursos académicos. Por ejemplo, el atributo url indica la ruta donde se ubica el bundle OSGI que contiene el material docente del curso.

Las relaciones (etiquetadas en XML con el tag *CardinalityBased_Relationships*) describen la transición entre un curso y sus cursos posteriores. Por ejemplo, entre el curso con Id 3 y el conjunto de cursos <4, 5, 6> existe una relación de tipo MFM:Alternative. Los atributos *LowerBound* y *UpperBound* indican que de este conjunto ha de seleccionarse como mínimo 1 curso y como máximo 2.

¹ MModeling Software Kitt: <http://www.moskitt.org/>

```
<Features Id="3">
  <Attributes Name="SubjectName"
    Vaue="Introduction to Graphs"/>
  <Attributes Name ="url"
    Vaue=http://myserver.com/course?id=1"/>
  <CardinalityBased_Relationships
    xsi:type="MFM:Alternative" LowerBound="1"
    UpperBound="2"/>
  <CardinalityBased_Relationships
    xsi:type="MFM:Optional" UpperBound="1"
    To="//@Features.4"
    From="//@Features.2/@CardinalityBased_Relationships.0"/>
  <CardinalityBased_Relationships
    xsi:type="MFM:Optional" UpperBound="1"
    To="//@Features.5"
    From="//@Features.2/@CardinalityBased_Relationships.0"/>
  <CardinalityBased_Relationships
    xsi:type="MFM:Optional" UpperBound="1"
    To="//@Features.6"
    From="//@Features.2/@CardinalityBased_Relationships.0"/>
</Features>
```

Figura 2. Serialización XML del Itinerario Formativo usando la sintaxis de un modelo de características.

La definición del itinerario formativo es el primer paso por parte del equipo pedagógico para obtener caminos personalizados a los estudiantes. El siguiente paso utiliza como entrada un itinerario académico, las preferencias de los estudiantes y sus resultados de años anteriores con el objetivo de aplicar algoritmos que calculen caminos personalizados. La próxima sección describe con detalle este proceso.

4 Adaptabilidad en el proceso de aprendizaje

El LMS seleccionará el curso siguiente en base a las calificaciones del usuario en sus cursos pasados recientemente así como en los caminos que más éxito han proporcionado al resto de usuarios. Este tipo de toma de decisiones basado en el comportamiento colectivo de sistemas descentralizados y auto-organizados se conoce como "inteligencia de enjambre", una expresión introducida en el contexto de la robótica por Gerardo Beni y Jing Wang [Beni y Wang, 1989] y que se inspira en el comportamiento social de los animales [Miller, 2007].

4.1 Optimización por colonias de hormigas

Para solucionar el problema del cálculo del mayor camino para cada usuario, teniendo en cuenta la retroalimentación proporcionada por el resto de usuarios del sistema, los miembros del enjambre, proponemos en este trabajo la aplicación de un algoritmo de optimización por colonia de hormigas [Dorigo et al., 1999].

En nuestro sistema, un alumno representa una hormiga moviéndose libremente entre los nodos del árbol de la Figura 1, liberando una cantidad variable de feromonas cada vez que supera un curso y se dispone a seleccionar el siguiente. La función que determina la cantidad de feromonas a depositar es una representación proporcional del éxito obtenido por el alumno en el curso recién evaluado, siendo por tanto esta cantidad de feromonas una medida del éxito obtenido.

Así, si tenemos en cuenta una escala de calificación de 0 a 10, el equipo pedagógico puede determinar como aceptable para cada curso una calificación determinada m , con $0 < m < 10$, y definir la cantidad de feromonas a depositar en el camino seguido hasta este curso en función de m como:

$$\Phi_{ij}(\varepsilon, m) = \begin{cases} -(1 - \varepsilon) \cdot \varphi - \omega_3 \frac{m - \varepsilon}{10}, & \text{if } \varepsilon < \frac{m}{10} \\ \varepsilon \cdot \varphi + \omega_3 \frac{\varepsilon - m}{10}, & \text{if } \varepsilon \geq \frac{m}{10} \end{cases}$$

dónde φ es una constante que representa el valor unitario de feromonas y ε es la calificación obtenida por el alumno en el último nodo evaluado, en la cual se encuentra. El rango de esta calificación se traslada a $[0, 1]$ dónde 0 representa la mínima calificación posible y 1 la máxima. Para potenciar el efecto de las últimas feromonas depositadas sobre el camino seguido por dicho alumno, los tres últimos arcos seguidos por este también verán incrementada la cantidad de feromonas existente en ellos, recibiendo el penúltimo la mitad de feromonas que el último, el antepenúltimo un tercio y el anterior una cuarta parte, tal y como otros autores proponen en este tipo de sistemas [Gutiérrez et al., 2007]. ω_3 es una constante de calibración del sistema que ayudará a estudiar y controlar el impacto de la calificación mínima requerida en la cantidad de feromonas depositadas. Por tanto, la cantidad de feromonas de un arco depende de la calificación de cada alumno que haya tomado ese arco y puede influir positivamente o negativamente. Evidentemente, si ningún alumno a escogido ese arco su cantidad de feromonas será nula.

Por otro lado, el equipo pedagógico puede asignar pesos a cada arco, para ponderar la importancia de unos caminos frente a otros. Este peso puede representar la complejidad pedagógica de la transición de un curso a otro, la importancia del curso destino, el esfuerzo o número de horas en relación con el total estimado del curso. Estos pesos son valores estáticos muy útiles para definir la función de ajuste que

proporcionará una medida representativa de la idoneidad del arco ij para el alumno, según sus necesidades.

$$f_{ij} = \omega_1 \cdot W_{ij} \cdot P_{ij} + \omega_2 \cdot \Phi_{ij}$$

Esta función será usada por el sistema para decidir el siguiente curso (nodo del árbol) que el alumno seguirá, construyendo dinámicamente y en tiempo de ejecución el camino de itinerario de aprendizaje del alumno.

En esta función de ajuste W_{ij} representa el peso pedagógico asociado al arco ij , Φ_{ij} es la cantidad de feromonas depositadas en el arco ij , y ω_1 y ω_2 son constantes de calibración del sistema. P_{ij} es el factor de idoneidad del arco ij , calculado *a priori* a partir de los resultados previos, actuando como multiplicador del peso pedagógico. Este factor podría ser proporcionado por el equipo pedagógico como un valor constante (por ejemplo la parte proporcional al número de nodos) o podría ser calculado dinámicamente tal y como se explica en [Márquez et al., 2008] usando Redes Bayesianas. El primer sumando en (2) es clave para evitar que el sistema quede bloqueado en un mínimo local causado por el efecto del segundo sumando, factor de las feromonas, tal y como demuestra en [Gutiérrez, 2007].

La función de ajuste puede ser calculada teniendo en cuenta la calificación (ε), la mínima deseada (m) y el tiempo (t), tenido en cuenta como iteraciones puntuales o acciones (la acción de superar un curso sería la unidad de medida). Teniendo en cuenta esto, la cantidad de feromonas a depositar en función del tiempo puede expresarse como:

$$\Phi_{ij}^t(\varepsilon, m) = \rho \Phi_{ij}^{t-1} + \lambda(i, j) \Phi_{ij}(\varepsilon, m)$$

En la fórmula anterior, ρ es la tasa de evaporación de las feromonas. Para evitar un impacto desmedido de las feromonas en la toma de decisiones del sistema, las feromonas, al igual que en la vida real, se evaporarán parcialmente con el paso del tiempo, perdiendo así intensidad y por tanto minimizando su impacto en el sistema. Puesto que los usuarios del sistema (los alumnos) pueden cerrar su sesión y abandonar el curso cuando deseen y reanudarlo cuando deseen desde el mismo punto en que lo dejaron, el tiempo, debe entenderse tal y como hemos comentado anteriormente como una sucesión de acciones o eventos que se producen al finalizar un curso. La tasa de evaporación es simplemente un factor positivo y menor que uno, que se aplicará a todos los arcos cada vez que un alumno seleccione un nuevo curso, disminuyendo el número de feromonas depositadas en él. Un valor de $\rho = 0,9$ es una tasa de evaporación típica. La función lambda devuelve 1 si el nodo fue el último nodo evaluado por el usuario y cero en caso contrario y sirve para asegurar que un usuario sólo

deposita feromonas directamente² en el último curso realizado. Considerando la tasa de evaporación, la cantidad de feromonas a depositar en un arco en un instante t es:

$$\Phi_{ij}^t(\varepsilon, m) = \left\{ \begin{array}{l} \rho \Phi_{ij}^{t-1} - (1-\varepsilon)\rho - \omega_3 \frac{m-\varepsilon}{10}, \text{ if } \varepsilon < \frac{m}{10} \\ \rho \Phi_{ij}^{t-1} + \varepsilon\rho + \omega_3 \frac{\varepsilon-m}{10}, \text{ if } \varepsilon \geq \frac{m}{10} \end{array} \right\}$$

que puede transformarse en:

$$\left\{ \begin{array}{l} \Phi_1^x + \rho^t (\Phi_0 - \Phi_1^x), \text{ if } \varepsilon < \frac{m}{10} \\ \Phi_2^x + \rho^t (\Phi_0 - \Phi_2^x), \text{ if } \varepsilon \geq \frac{m}{10} \end{array} \right\}$$

dónde $\Phi_1^x = \frac{-(1-\varepsilon)\rho - \omega_3 \frac{m-\varepsilon}{10}}{1-\rho}$

y $\Phi_2^x = \frac{\varepsilon\rho + \omega_3 \frac{m-\varepsilon}{10}}{1-\rho}$

y Φ_0 es el valor inicial. Nótese que si $|\rho| < 1$ entonces:

$$\lim_{t \rightarrow \infty} \Phi_{ij}^t = \left\{ \begin{array}{l} \Phi_1^x, \text{ if } \varepsilon < \frac{m}{10} \\ \Phi_2^x, \text{ if } \varepsilon \geq \frac{m}{10} \end{array} \right\}, \text{ esto es, } \Phi_i^x \text{ es}$$

estable.

4.2 Aplicación de las redes Bayesianas al cálculo del factor de idoneidad

Para calcular el factor de idoneidad P_{ij} del arco ij , se usará una red Bayesianas. Este factor determinará la probabilidad *a priori* de tener éxito tomando el arco ij . Puede usarse para diagnosticar las causas de fracaso escolar [] pero también para elegir un arco que refuerce el bajo conocimiento adquirido por el alumno hasta entonces. Esto es, para potenciar caminos alternativos.

² Recuérdese que el sistema se encarga de depositar otras cantidades menores en los tres nodos anteriores para reforzar el camino. Estas cantidades son proporcionales a la calculada.

En una red Bayesianas de razonamiento predictivo, el conocimiento se propaga desde las causas (evidencias) hacia los efectos. Las calificaciones de los alumnos en los cursos finalizados hasta el momento son las únicas posibles causas, aunque sería posible definir otras evidencias (factores socioeconómicos del entorno familiar del alumno, pautas de comportamiento, participación, expresión escrita, etc) que serían introducidas por el profesor para conseguir una personalización más compleja. En este trabajo, por simplicidad, consideraremos el conocimiento asimilado por los alumnos medido a través de sus respectivas calificaciones como la única causa de éxito o fracaso que influirá en la decisión del siguiente curso.

Podemos representar una red Bayesianas como un grafo dirigido sin ciclos que muestre las dependencias cognitivas de cada curso.

Cada nodo representa una variable discreta con un número finito de estados y los arcos representan la dependencia causal entre variables. Cada nodo estará asociado a una tabla de distribución de probabilidad condicional sobre los estados de una variable. Este grafo no tiene por qué ser el mismo que el definido por el equipo pedagógico, y tampoco tiene por qué conexo. En este grafo dos cursos consecutivos pueden no depender entre sí y sí tener dependencias con uno o varios cursos no directamente consecutivos. Incluso es posible que un curso no dependa de ningún otro, en cuyo caso las probabilidades de obtener un determinado rendimiento de entre un conjunto discreto de niveles no son en absoluto condicionadas, y pueden estiliarse a un valor igual para todos los niveles.

Estamos suponiendo que el rendimiento académico de cada alumno en un curso depende únicamente de otros cursos previos, premisa que no es absolutamente cierta [Brown y VanLehn, 1980] porque depende de muchos otros factores. Brevemente, si los conceptos de un curso C dependen de los conceptos explicados en otros dos cursos, A y B (diremos que C depende pedagógicamente de A y B) la calificación esperada para el curso C depende, a priori, de las calificaciones que el alumno haya obtenido en los cursos A y B, que constituyen las evidencias.

Los datos históricos de cursos anteriores ayudarán a construir las probabilidades iniciales para una población de alumnos similar en la red Bayesianas. Las calificaciones se discretizarán a tres niveles: BAJA, MEDIA y ALTA, cuyos umbrales deben ser definidos por el equipo pedagógico. Un ejemplo de estas probabilidades puede verse en la Figura 3. En ella se muestra la probabilidad de obtener unas calificaciones según estos tres niveles en el nodo C, condicionada a las obtenidas en los nodos A y B.

A	
High	10,0%
Medium	60,0%
Low	30,0%

B	
High	15,0%
Medium	57,0%
Low	28,0%

C									
A	High			Medium			Low		
B	High	Medium	Low	High	Medium	Low	High	Medium	Low
High	65,0%	55,0%	25,0%	67,0%	26,0%	17,0%	28,0%	18,0%	11,0%
Medium	28,0%	35,0%	53,0%	23,0%	55,0%	45,0%	55,0%	43,0%	39,0%
Low	7,0%	10,0%	22,0%	10,0%	19,0%	38,0%	17,0%	39,0%	50,0%

Figura 3. Tablas de probabilidad condicionada

La clave de este desarrollo reside en el Teorema de Bayes que afirma que relaciona la probabilidad condicional y marginal de dos eventos estocásticos. Así, la probabilidad de la hipótesis h conociendo la evidencia e es igual a la probabilidad de la evidencia, conocida la hipótesis, multiplicada por la probabilidad de esa hipótesis y dividida por la probabilidad de la evidencia:

$$P(h/e) = \frac{P(e/h) \cdot P(h)}{P(e)}$$

En esta red, las calificaciones de los cursos previos son las evidencias. En aquellos nodos en los que es posible elegir entre varios caminos alternativos, es especialmente importante calcular el factor de idoneidad de cada nodo candidato al siguiente curso, que puede ser determinado seleccionando el curso que mejor se ajusta a la restricción buscada. Por ejemplo: que la suma de las probabilidades de obtener una calificación ALTA o MEDIA sea mayor del 65%, que se ofrezca una probabilidad de obtener una calificación MEDIA superior al 50%, etc.

Esta tarea consiste en calcular los π -valores [Gutiérrez et al., 2007] que informan acerca de los valores del nodo X más probables, basados en la evidencia relacionada con las causas de X (según la evidencia por encima de X). Estas ecuaciones son calculadas directamente cuando la red es simple, pero en una red compleja o con un número suficientemente amplio de nodos es preferible usar algoritmos de propagación de la probabilidad y métodos de aproximación (Pearl, Joint Computation, simulación estocástica, Simple Bayes, K2, Clustering, etc) [Li y D'Ambrosio, 1994]

Un ejemplo de función de idoneidad para el arco ij podría ser:

$$P_{ij} = \frac{P(J_{High} / I) \cup P(J_{Medium} / I)}{0.33} [0.7, 1]$$

Esta función calcula el factor de idoneidad para el arco ij como la suma de las probabilidades de obtener una calificación media o alta en el curso del nodo j conociendo la probabilidad en el nodo origen i, cuya nota es conocida, si existe dependencia entre los nodos i y j. Esta probabilidad se establece constante en 0,33 si no existe dependencia del nodo j con el nodo i.

5 Conclusión y trabajo futuro

En este trabajo amplia otros trabajos anteriores y presenta la aplicación del modelado de características y gestión de la variabilidad para describir y representar una programación didáctica en un sistema de aprendizaje orientado por competencias en el que se aplican técnicas adaptativas para personalizar el itinerario formativo a las necesidades de los alumnos.

El trabajo futuro se centrará en el estudio y desarrollo de estas técnicas adaptativas así como en la aplicación de herramientas que permitan un fácil diseño y manipulación de las programaciones didácticas por parte del equipo pedagógico.

Referencias

- [Batory, 2006] Don Batory. *Multilevel models in model-driven engineering, product lines, and metaprogramming*. IBM Syst. J., IBM Corp., 2006, 45, 527-539.
- [Beni y Wang, 1989] Gerardo Beni y Jing Wang. Swarm Intelligence in Cellular Robotic Systems. *In Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy. Junio 26-30, 1989.
- [Brown y VanLehn, 1980] J.L. Brown, K. VanLehn. Repair Theory: a generative theory of bugs in procedural skills. *Cognitive Science*. Vol. 4, pags. 379-426. 1980.
- [Czarnecki et al., 2004] Krzysztof Czarnecki, Simon Helsen y Ulrich Eisenecker. Staged Configuration Using Feature Models. *Software Product Lines, 2004*, 266-283.
- [Dorigo et al., 1999] Marco Dorigo, Gianni Di Caro, Luca M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5 (2): 137-172, 1999.
- [Gutiérrez, 2007] Sergio Gutiérrez. *PhD Thesis: Secuenciamiento de actividades educativas orientado a la reutilización y la auto-organización en tutoría inteligente*. Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.
- [Gutiérrez et al., 2007] Sergio Gutiérrez, Grégory Valigiani, Yannick Jamont, Pierre Collet, Carlos Delgado Kloos. A Swarm Approach for Automatic Auditing of Pedagogic-

- al Planning. In *Proceedings of Seventh IEEE International Conference on Advanced Learning Technologies*, 2007. ICALT 2007. 18-20 July 2007 Page(s):136 – 138
- [Li y D'Ambrosio, 1994] Zhaoyu Li, Bruce D'Ambrosio. Efficient inference in Bayes networks as a combinatorial optimization problem. *Int. J. Approx. Reasoning* 11(1): 55-81. 1994
- [Márquez, 2007] José Manuel Márquez. *Estado del arte del eLearning. Ideas para la definición de una plataforma universal*. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Junio 2007.
- [Márquez et al., 2008] José Manuel Márquez, Juan Antonio Ortega, Luis González-Abril y Francisco Velasco. Creating adaptive learning paths using Ant colony Optimization and Bayesian Networks. *International Joint Conference on Neural Networks (IJCNN). IEEE World Congress on Computational Intelligence*. Hong Kong, June 1-6, 2008.
- [Miller, 2007] Peter Miller. Swarm Theory. *National Geographic Magazine*. July 2007.
- [Nkambou y Tchétagni, 2004] R. Nkambou, J.P.M. Tchétagni. Diagnosing Student Errors in E-Learning Environment using MPE Theory. In *Proceedings of Web-based Education*. Innsbruck, Austria. 16-18 February 2004.
- [Rey-López et al., 2006] Marta Rey-López, Ana Fernández-Vilas, Rebeca Díaz-Redondo y José Pazos-Arias. Providing SCORM with adaptivity. In *Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 981-982
- [Ruiz et al., 2004] José Luis Ruiz, José Luis Arciniegas, Rodrigo Cerón, Jesús Bermejo y Juan Carlos Dueñas. A Framework for Resolution of Deployment Dependencies in Java-Enabled Service Gateways. *Scientific Engineering of Distributed Java Applications. Lecture Notes in Computer Science*. Volume 2952/2004, Page(s) 1-11. March 2004.