Classifying a new teacher

After the new teacher registers, the system takes his data and classifies the teacher in one of the existing categories (Beginner, Intermediate, Advanced) by means of a Bayesian classifier, a common technique in user modeling. Our system uses a naïve Bayesian classifier to classify teacher's characteristics and assigns the new teacher to the category in which the teacher's data are most suitable. Once the teacher belongs to a category, the most frequently used class of examples in this category is used to recommend sessions and examples to the new teacher, as explained in section 5.4.

Naïve Bayes classifiers have been used and recommended in adaptive hypermedia (Webb *et al.*, 2001; Miyahara and Pazzani, 2002) because they reduce the latency between requests and answers via Internet. Furthermore, a Bayesian classifier does not compare the teacher's data with cases in the database as the K-NN classifier would. Instead of this, a Bayesian classifier keeps a summarized image about the frequency with which different teacher's characteristics appear in the database. This kind of informed search reduces the time needed to find the most suitable category or class for a teacher.

As explained in section 2.2.7., the Bayesian classifier is a *supervised learning* method that requires a training phase. Synthesized data was used for the training phase because there are not enough teachers teaching our curriculum. Using synthesized data (see Table 4.6.) is not an uncommon practice when constructing complete systems (Herlocker *et al*., 2004) that apply collaborative filtering. Each of our categories (beginner, intermediate and advanced) stores two types of teacher data, *Demographic* and *Background knowledge*. Demographic data is used because there is evidence of the relation between age and gender and the preferences of university teacher to use information technology (Cuneo *et al*., 2002). The data in both types is detailed as follows (we associate each characteristic with a letter for future identification):

Demographic

A. Age.

B. Gender.

Background

C. Teacher's self-evaluation of his skill or experience teaching the selected subject.

D. Number of times the teacher has taught this type of course.

E. Teacher's background discipline or area of expertise.

F. Teacher's skills using computers.

G. Teacher's experience using Web-based technology for teaching.

H. Teacher's experience using this Web site.

An example of a new teacher to be classified can be:

Age = 45

Gender = Male (M)

Self-evaluation = Beginner (B)

Times teaching this course = 1

Teacher's background = Network design (ND)

Teacher's skills = Medium (M) average

Teacher's experience using Web-based technology = Never (N)

Teacher's experience using this Web site = Never (N)

Generally, people until 30 years have the highest use of Internet, after that the use of information technology becomes lower with the pass of the time; finally, the period over 55 years age is of very low Internet use, for example (Cuneo, 2002). We have defined these rules that organize data about age in three ranges:

If teacher's age 30, then his age range is 1.

If teacher's age > 30 AND 55, then his age range is 2.

If teacher's age > 55, then his age range is 3.

This new teacher's characteristics are summarized as follows in Table 4.5.:

Table 4.5. A new teacher's attributes and values**.**

	Teachers' Characteristics								
Teachers	A	B	C	D	E	F	G	H	Class
A new teacher *E''*	2	M	B	1	ND	M	N	N	?

This example has to be classified in one of the three categories. A set of training examples is shown in Table 4.6., and Table 4.7. includes the values that the system uses for the different attributes.

Table 4.6. A training dataset.

A	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***Class***
3	F	I	1	ND	A	N	S	Beginner
3	NA	I	2	DM	H	S	S	Beginner
3	M	B	2	ND	L	N	S	Beginner
3	M	I	1	DM	A	S	N	Beginner
3	F	B	2	DM	A	S	O	Beginner
3	NA	I	3	DM	L	O	N	Beginner
3	M	B	1	DM	A	O	N	Beginner
3	F	B	3	O	L	O	O	Beginner
3	F	I	1	ND	H	O	S	Beginner
1	M	I	3	ND	H	S	S	Intermediate
1	NA	I	1	O	H	N	O	Intermediate
1	NA	B	2	O	A	O	N	Intermediate
1	M	B	3	ND	A	O	O	Intermediate
1	NA	B	3	O	H	O	S	Intermediate
1	N	A	1	O	A	S	O	Intermediate
2	F	I	3	O	H	S	O	Advanced
2	NA	I	2	ND	L	N	S	Advanced
2	F	A	3	DM	L	O	N	Advanced
2	F	A	1	O	H	S	S	Advanced
2	M	I	2	ND	H	S	S	Advanced

Table 4.7. Possible values for the different attributes**.**

A	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***Class***
1	F	B	1	ND	L	N	N	Beginner
2	M	I	2	DM	A	S	S	Intermediate
3	N	A	3	O	H	O	O	Advanced

The meaning of the values in Table 4.7. are

For age

1= teacher's age 30

2= teacher's age > 30 AND 55

3= teacher's age > 55

For gender

F= female

M= male

NA= not available

For teacher's self-evaluation

B= beginner

I= intermediate

A= advanced

For times teaching a course

1= never

2= 1 to 5 times

3= more than 5 times

For teacher's background discipline

DM= decision-making

ND= network design

O= other

For teacher's skills using computers

L= low

A= average

H= high

For teacher's experience using Web-based technology

N= never

S= sometimes

O= often

For teacher's experience using the Web site

N= never

S= sometimes

O= often

With these training dataset, it is possible to apply equation (11)

$$P(a_i|v_j)=\frac{nc+mp}{n+m} \tag{11}$$

where:

- a_j is the value of each attribute a_i tested for each class v_j. For example, an instance of a_i can be 2.
- n is the total number of instances in each class v_j, In this case of class Beginner, $n = 9$.
- n_c is the number of instances with attribute a_i and class v_j. This is the number of times that the value of the attribute a_i appears in each class v_j. For example, the value "M" appears one time in the Advanced category, in this case $n_c = 1$.
- p is a priori estimate for $P(a_i|v_j)$. We assume that the probabilities of all attributes are equiprobable and each attribute can have three possible values and $p = 1/3$.
- m is a constant used to avoid the possible consequences that could arise if $n_c = 0$ (in this case the calculation would be zero). We use m = 8 as a constant because we have eight attributes in the examples of our training dataset.

After the calculation of the probabilities of each attribute, we apply equation (10) adapted to this example:

$$v(E'')=\frac{\text{argmax}}{v_j\in\{B,I,A\}} P(v_j)\prod_i P(a_i|v_j) \tag{10}$$

the result is as follows:

Class	Probability
Beginner	0,0002572241
Intermediate	0,0000351170
Advanced	0,0000061686

Consequently, the new teacher is classified in the category "Beginner" because the classifier predicts that this class has the highest probability for this teacher. Once the teacher is classified, the system has new information with which to support the teacher with help and recommend sessions and examples.

Plans

A fundamental aspect for the recommentation of examples and the adaptive help is the recognition of the plan that the teacher must perform and the corresponding plan representation. We apply a Keyhole plan recognition (Wærn and Stenborg, 1995) that assumes teachers are indifferent to the plan-recognition process and requires less complex recognition mechanisms. Our representation of a teacher's plan in the Teacher Model allows the system to find appropriate help associated with each field in a Web page. In our plan representation, each teacher's plan and each activity, task and step are represented in the Teacher Model by a simple code made up of an eight-character string. We structured this code as follows: two characters represent the Curriculum, two characters are for the activity, the next character is for the task, and the following for the step, and the last two represent a field. Examples of the representation of the Curriculum of Network Design are shown in Table. 4.8. These examples detail the representation of a Network Design course as 01000000; the Reliability activity of this course as 01010000; a step of a task of this activity as 01011100; and, the first field of this step is represented by 01011101.

Table 4.8. The representation of a plan.

Course	*Activity*	*Step*	*Field*	*Meaning*
01	00	00	00	Network (Course 01 is not a specific Web page)
01	00	10	01	Session number 1 (Step 10 is used for Session numbers)
01	00	01	00	Session Step (Step 01 is for Session selection page)
01	00	01	01	Field 1 in Session Web page
01	01	00	00	Reliability activity (it is an activity, not a Web page)
01	01	11	00	Reliability activity, Task1, Step1 (just a Web page)
01	01	11	01	Reliability activity, Task1, Step1, Field1 (a field)

Each *eight-character code* identifies a plan or part of a plan and allows the system to store data related to each activity, task, step, and field (Rodríguez *et al.*, 2004b). The code identifies a record in the table of a database, which includes other data associated with this identification. For example, the code allows the system to identify each field of each page and to store statistics on the accesses to the help in each field. The date of creation or modification of a step can also be stored. When the teacher is following a sequence of steps to create a learning session, this plan representation allows the system to know the last step that the teacher authored to define the next step for the authoring tool. Information related to the next step supports the selection of help resources. If the teacher quits the sequence and jumps to other step, the system will use the plan representation of this new step to make decisions on recommending examples and selecting help. To recommend examples according to the current task (step), the generator and the recommender also take into account the information of the plan.

Retrieving and assigning resources

Once the teacher enters a Web page, the teacher's skills and plan representation (code) are in memory. The first five characters of this code show the context (course, activity and task) of the current step. The teacher's skill determines the type of help strategy and technique to use. Once the plan, the teacher's skills, and the help technique are defined, the next phase is filtering help resources to retrieve the appropriate resource for each help

technique (Rodríguez *et al.*, 2004b). These resources can be files with texts, graphics, videos, applications or simulations. Each resource file is indexed by a set of pairs *attribute/value*; these attributes are *complexity*, *media*, *language*, *load*, *help techniques*, *context area*, and *display mechanism*, and some of their possible values are detailed in Table 5.12.

Table 5.12. Attributes used to index resources

Attributes	**Values**		
Help Stategies	Basic help	Guidance	Assistance
Language	English	Spanish	French
Load	Low bandwidth	High bandwidth	
Area content	Usage	Domain	
Media	Text	Image	Multimedia
Complexity	Beginner	Intermediate	Advanced
Display mechanism	Form	Layer	Applet

Based on the three-dimensional model of the teacher's characteristics, the system defines the resources that are more suitable according to the teacher's skills, plan, help strategy, and preferences. This definition consists of retrieving the values of each teacher's characteristics that are stored in the Teacher Model, for example, his skills, plan, language, and preferred media to see messages. After retrieving a set of values from the Teacher Model, the system chooses the resources whose attributes match these criteria. By matching these values to the attributes of the resources, the system defines the appropriate resources for the step that the teacher is going to perform. For example, a resource such as an image can be selected because its attributes match the values of the teacher's characteristics (Rodríguez *et al.*, 2003). Using this same method (filter), the system also predicts the resources that might be needed later in a Web page. With the resources identified, the system can retrieve the corresponding resources and thus build a Web page linking resources to fields in the template. Some predicted resources are not used immediately and the Web page loads them for future use with help techniques.

To link a resource to a specific field in a Web page, the resources (texts, images, multimedia files) are indexed by hand in a many-to-many way to fields in different Web pages. This process is tedious and time consuming but highly precise. The reasons to use

this method are that other methods make more mistakes choosing resources (Leung, 2001). In addition, resources, especially audiovisual ones such images, sounds, video must be indexed to identify them according to the pairs attribute/value. A table in a database stores these relationships to enable the system to assign resources to fields in templates while a Web page is being created. The filtering process of resources is depicted in Figure 5.7.

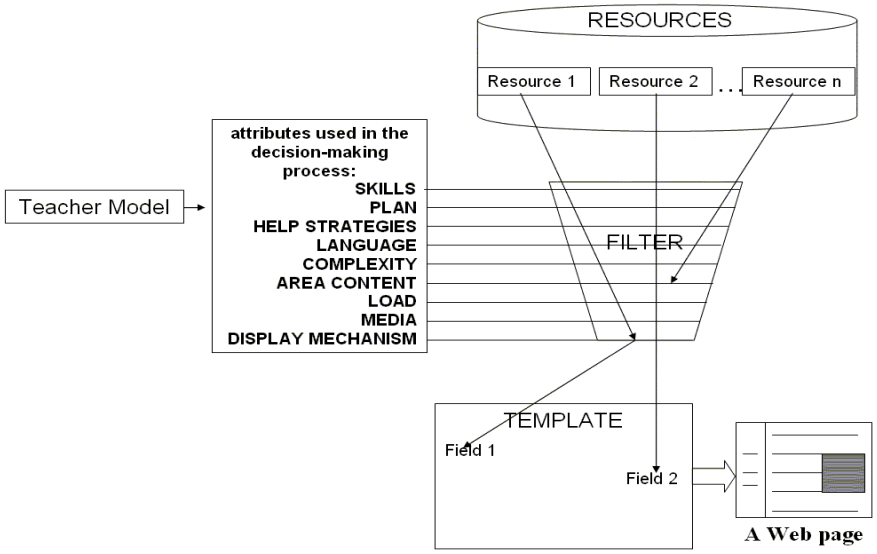

Figure 5.7. The system uses criteria to make decisions about which resources are located in a Web page (Rodríguez *et al*., 2004b).

The advantage of this method is the flexibility to reuse resources and to build Web-pages dynamically according to multiple attributes that acts as a filter. Most systems do not take into account additional information about personal characteristics to adapt help content and select the corresponding resources. Usually, Web-based help systems are not knowledge-based and they only provide context-sensitive help regarding the task that a user is performing at that moment. Choosing resources based on information stored in the Teacher Model and predicting the help contents and resources that a teacher might need later are a new vision of adaptation in help systems, particularly Web-based help systems.