Université de Montréal

C.A.M.E.L.E.O.

A Cultural Adaptation Methodology for E-Learning Environment Optimization

par

RYAD ADEBOLA RAZAKI

UNIVERSITE DE MONTRÉAL

DÉPARTEMENT d'INFORMATIQUE et de RECHERCHE OPERATIONELLE

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès Sciences
en Informatique

septembre, 2007

Université de Montréal

Faculté des études supérieures

Cette mémoire intitulée :

C.A.M.E.L.E.O.:

A Cultural Adaptation Methodology

for E-Learning Environment Optimization

présentée par :

Ryad Adebola Razaki

a été évalué par un jury composé des personnes suivantes :

Jian-Yun Nie, président-rapporteur

Claude Frasson, directeur de recherche

Nadia El-Mabrouk, membre du jury

Mémoire accepté le 20 novembre 2007

Résumé

Les méthodes pédagogiques varient énormément de part le monde. Un étudiant d'un village en Corée apprend d'une façon différente d'un étudiant d'un village en Slovaquie. Cette variabilité n'est pas surprenante et peut être liée à leur environnement immédiat. Ceci signifie que la même information, lorsque présentée à des groupes d'utilisateurs culturellement différents, peut mener à des interprétations différentes et ainsi causer la conception d'idées fausses sérieuses si elle n'est pas adaptée aux spécificités de cette population. Ceci est particulièrement vrai dans le cas où l'information est disponible sur Internet et accessible directement aux étudiants au sein de leur propre environnement. Ainsi, donner un accès mondial à la même connaissance peut générer des idées fausses, non du fait de l'information en elle-même mais plutôt si aucune attention n'est portée à la manière dont elle est présentée. Ceci parce qu'il y a des règles qui définissent la manière dont chaque culture enseigne. Cependant, bien que la recherche prouve que "le manque d'adaptation culturelle est la raison principale pour laquelle le e-learning ne fonctionne pas pour une audience globalement distribuée", les Systèmes Tutoriels Intelligents (STI) ne sont pas adaptés à la culture. Dans cette thèse nous présenterons une méthodologie culturelle d'adaptation pour l'optimisation de l'environnement d'e-learning (CAMELEO) dont le but est de fournir des moyens pour les STI de s'adapter à la culture des étudiants. Ceci exigera l'ajout au modèle de l'apprenant actuel d'une extension pour la caractérisation culturelle d'un étudiant: le modèle culturel de l'apprenant (CSM), obtenu avec l'utilisation des techniques de filtrage collaboratif.

Mots-clés: Culture, Système Tutoriel Intelligent, e-learning, filtrage collaboratif, adaptation culturelle, système multi-agent

Abstract

Pedagogical methods vary enormously around the world. Students from villages in Korea learn in a different manner from students from villages in Slovakia. This variability is expected and can be related to their immediate environment. This means that the same information presented to culturally different groups of users can lead to different interpretations and yield serious misunderstandings if it is not adapted to the specificities of that population. This is especially true in the case where the information is available on the Internet and accessible directly to students within their own environment. So, giving a worldwide access to the same knowledge may lead to some misinterpretations not because of the information in itself but rather if no caution is put in the way in which it is presented. This is because there are rules that define the way to teach in each culture. Yet, even though research shows that "the lack of cultural adaptation is a leading reason why e-learning fails to work for a globally distributed audience", Intelligent Tutoring Systems (ITS) are not culturally adapted. In this thesis we will introduce a Cultural Adaptation Methodology for E-learning Environment Optimization (CAMELEO) whose purpose is to provide means for ITS to adapt to learners' cultural background. This will require the extension of the current student model with a new feature for cultural characterization of a student: the Cultural Student Model (CSM), generated with the use of collaborative filtering techniques.

Keywords: Culture, Intelligent Tutoring System, e-learning, collaborative filtering, cultural adaptation, multi-agent system

Table of contents

List of tables

List of figures

To my parents, friends and family

Acknowledgements

First and foremost, I would like to thank my research advisor, Claude Frasson, for his guidance, but also for his constant support as well as his unflinching understanding. I would also like to thank the Heron Lab team, especially Emmanuel Blanchard for is collaboration and for providing me with a strong direction and key clues for pursuing my research efficiently and Alicia Heraz for her moral support and our enriching discussions.

I would like to thank everyone around the world that took part in my experimentations for the time and consideration they gave me and the advice they provided me with to give the best possible orientation to my work.

I would like to thank the staff and faculty of the DIRO and in particular Mariette Paradis, for undergoing the stress that I so often put them through.

Finally, I would like to thank my parent for giving me the opportunity to follow my path in life.

Introduction

Goals of the thesis

A great deal of progress has been made these past few decades in the field of *Intelligent Tutoring Systems (ITS)*. Computer assisted teaching has evolved from simple tutoring software to the more complex ITS architectures [Burns and Capps, 1988]. ITS architectures are generally divided into three mains parts:

- A first part represents the knowledge to be taught. It is known as the *knowledge base*.
- A second part takes into account various aspects of the learners' profiles, in order to better target the knowledge that is missing or incomplete. That part is known as the *student model*. It is aimed at being the best possible representation of the learner.
- A third part known as the *Tutor* contains strategies for teaching material. The strategies vary according to both the knowledge to be taught and also more and more according to the learner that is being taught.

While that structure works perfectly well in most cases, gradually, inefficiencies have started to emerge, mostly concerning the Student Model and its use by the Tutor. This is especially true when the same teaching material is available to students from different origins. Indeed, pedagogical methods around the world vary enormously. Because of that students are used to being presented material in different manners. They have developed different methods for understanding concepts that are being communicated to them. Therefore, students from villages in Africa learn in a different manner from students from villages in Asia. That variation is expected and can be related to their immediate environment. This means that the same information when presented to different groups of users can lead to different interpretations and thus yield serious misunderstandings. This can be particularly problematic on the Internet where the information is available and accessible directly to students within their own environment; an environment that can be drastically different from that of the initial target group of the information.

Giving a worldwide access to the same knowledge may lead to some misinterpretations not because of the information in itself but because of the way in which it is presented. This is because there are principles, difficult to formulate, that define the way to teach in each culture. We know that those principles exist, even if they are intangible, because teachers around the world are trained differently according to the population they are trying to teach. Indeed, studies have shown that "the academic achievement of ethnically diverse students will improve when they are taught through their own cultural and experiential filters" [Au & Kawakami, 1994; Gay, 2000]. In other words, teaching strategies must be based on culture and the strategy that is closest to a students "cultural filter" should be recommended to teach that student. However, those cultural rules for teaching vary enormously and many of them are subjective or implicit. Besides, some of those rules require a very close acquaintance with the student. Some teachers base their relationship with students on his family background; Some base it on other factors; But all rely on the student's reactions and responses. In order to convey the same meaning, a teacher would rely on different strategies for students of different backgrounds.

The issue is therefore, in the case of ITS, to present the same information to culturally different populations across the world while expecting to transmit the same understanding. Research shows that "the lack of cultural adaptation is a leading reason why e-learning fails to work for a globally distributed audience" [Dunn and Marinetti, 2004]. The problem is to find the way to teach a student while considering his cultural background. Yet, as of today, student models are missing any form of cultural characterization. Before we can adapt teaching according to a student's culture, we need to add elements of culture into the student model. This first hint of a solution brings up one other much bigger issue. How does one determine the "cultural background of a user"?

Some efforts have already been made towards integrating a cultural aspect to ITS in the past. In particular, the system known as the Culturally AWAre System (CAWAS) [Blanchard and Frasson, 2005] that generates a cultural profile for students on the basis of the Hofstede's system of values [Hofstede, 2001] which represent national cultures with a set of dimensions and associated scores. This is a rule-based system that makes decisions based on rules that are triggered on the basis of the user's national culture.

But, that type of system is limited insofar as it fails to take into account a "more contemporary view of culture. Indeed the former distinctions solely based on geographical distance are growing obsolete, leaving way to less tangible characteristics such as social class and level of education or level of mobility" [Urry 2002]. This is to say that culture is more complex than only the geographical origin of an individual. There are definitions of culture that encompass various other factors, such as social, professional background. Indeed, people of different origin might feel closer to one another than they do to other people from their own countries, because of similarity of social standing or because they have access to the same media. One needs to explore those more exhaustive definitions of culture in order to exactly determine what brings some people into the same group, the same cluster.

Culture has been defined as "a process of production and reproduction of meanings in particular actors' concrete practices (or actions or activities) in particular contexts in time and space" [Kashima, 2000]. This is same as to say, in the case of ITS, that an individual can be understood to belong to the same culture as a learner if that individual has been proved to respond in a similar manner when presented with the same situation as the learner. This implies that determining a learner's cultural profile is the result of the process of observing his behavior or interaction, his response to the system. Taking that definition of culture as a basis for constructing a cultural profile means that a system in order to apply a strategy must beforehand verify that the strategy has worked on users that are similar in their previous behavior to the learner. This is in accordance to the definition of culture as a product of human behavior. Culture is not tangible; it is simply the similarity between people's preferences. Two individuals should belong to the same culture if they react similarly to a suggested strategy. Also, similar reactions to suggested strategies brin g together individuals, whereas dissimilarities of reactions are an indication of diverging cultures. As far as ITS are concerned, a particular strategy is recommended for a learner if it has yielded good result for an individual that is similar to the learner. This introduces the idea of recommending strategies and verifying their efficiency in order to confirm or infirm the system's knowledge of a learner. There is a huge area of knowledge retrieval and management that deals with the problem of building profiles as a result of recommendations. This is a hint that we need to turn toward *recommendation systems* in order to find the most efficient way to build a cultural profile. Recommendation system are helpful in this case because the characteristics of culture are not clear and recommendation system do not always require clearly formulated specification of characteristics before they can function.

The main goal of this thesis is to determine a cultural model that can be included in current student models to take care of learners' cultural specificities.

Organization of the thesis

The scope of this thesis is rather vast because it touches various domains: first ITS, already cited and which is our main area of study, second student models which we believe are lacking in some features, and finally recommendation systems which will help us in building those missing features.

In Chapter 1 we will survey the advances made in those domains. First, in the domains of ITS, we will review the various design patterns under which ITS architectures have been classified. We will explain the purpose and specificities of each pattern. Then, in the domain of Student Modeling, we will follow the evolution of student models from simple cognitive student models to psychological, emotional and finally to the first steps taken in building a cultural student model. We will illustrate with practical examples each of those models. Finally, in the domain of recommendation systems, we will discuss in length the concepts of content-based filtering and collaborative filtering. The purpose of studying those filtering methods is to determine the most efficient way to create clusters of similar user profiles, in our case cultural profiles, based on their interaction with a system.

In Chapter 2, we will propose a solution to the problem of cultural adaptation in an e-learning environment. In order to reach that solution, we will first discuss theories from various areas of science (mainly psychology, social and pedagogical) that give a clear definition of culture and give leads on cross-cultural teaching methods. Those are the theories on which our solution will be based. So, on the basis of the findings of those sciences, we will suggest our solution: the Methodology for Cultural Adaptation in an E-Learning Environment (C.A.M.E.L.E.O.).

Next, in Chapter 3 we will present a possible architecture for a system that follows the semantics of the methodology introduced in Chapter 2. It is a multi-agent system named CAMELEO. We will start by giving a justification for the choices made concerning that architecture. Then we will give a detailed description of every module involved in that architecture. Finally we will describe in length the main functionalities provided by the architecture.

Chapter 4 is the description of a system that we built according to that architecture. It is an implementation of the CAMELEO architecture. The focus in this section will not only be on describing this particular implementation, but it will also be on the ways in which the implementation relates to the theories mentioned in Chapter 2. We will therefore justify that our implementation is truly an answer to the problem at hand.

Chapter 5 presents an experimentation made using the implementation of CAMELEO described in Chapter 4. We will discuss results obtained from that experimentation and see whether or not CAMELEO holds its promises, meaning whether we have succeeded at building a system that takes into account the culture of a learner in an e-learning environment.

Finally, and in Conclusion, we will mostly recapitulate over the steps that we went through building CAMELEO and verify whether or not we reached our goal, which is again, to determine a cultural model that can be included in current student models to take care of learners' cultural specificities. We will also discuss the limitation of our system and mention some future works regarding that project. That work will mostly be concerned with answering those limitations. Finally we will show the benefits of extending that methodology to areas outside of ITS.

Chapter 1: Literature review

1.1. Intelligent Tutoring Systems

An Intelligent Tutoring System (ITS) tries to model a student's knowledge, style and preferences in order to help navigate them through the learning process in an individualized manner, one that meets their needs. It provides appropriate help, suggests the next step in the learning cycle and presents material in a way that matches their preferences [Kelly and Tangney, 2002].

ITS architecture designs have been classified into patterns that follow software designs logic. Some of the main patterns that have been discovered are the *Classic ITS architecture*, the *Intelligent Computer Supported Collaborative Learning (ICSCL) Model* for collaborative learning systems, the *Generalized Pedagogical Agent* (GPA) and the *Co-Learner* pattern [Devedzic, 2005].

1.1.1. Classic ITS architecture

The Classic ITS Architecture (Figure 1.1) defines its modules according to the role they play in the tutoring process. There are four modules: the *Expert Module*, the *Student Module*, the *Tutor Module* and the *Communication Module*.

- The *Expert Module* represents the domain knowledge. It describes the ability to solve all problems that could arise in that particular domain.
- The *Student Module* manages the student model. It contains information about the student's knowledge. The student model will be described in more details in section 1.2.
- The *Tutor Module* represents the pedagogical competences. It manages the various tutoring and institutional strategies available to the system.
- The *Communication Module* describes the interface between the student and the learning environment. The module contains information about the means of interactions between the student and the learning environment.

A typical instance of one of the many ways in which those modules interact is as follows: The tutor decides of an appropriate task according to the Student Module; The Communication module interprets the task and the student's choice of solution for the Expert module. The Expert module observes the student problem solving behavior and makes a diagnosis. This diagnosis is used to update the Student Module, and so on.

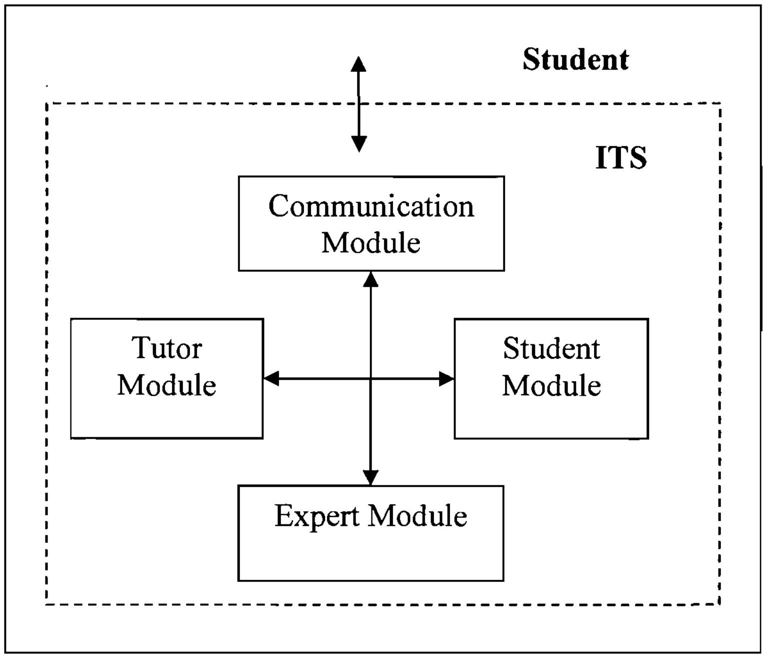


Figure 1.1 Classic ITS architecture (system border in dotted lines) [Wenger, 1987]

The Classic architecture describes the communication between a single learner and the learning environment. Yet very often, and especially in ITS, learning happens within a group. In such cases, the learning environment encompasses not only the tutoring system, but also the other learners in the ITS. This is because information can be provided by the Expert Module as well as by using the knowledge of other users or group of users. That is the situation of collaborative learning. How can that situation be modeled?

1.1.2. ITS architecture for collaborative learning

In a collaborative learning situation, it can be necessary to keep track of the knowledge not only of single individuals, but of particular groups. In this case, the group itself becomes a learner. Techniques used in ITS for single learners, such as adapting the learning material to the needs of the groups or specific members, have to model the group as well as the individual learner [Devedzic, 2005]. This is done using an Intelligent Computer Supported Collaborative Learning (ICSCL) Model. That model requires the system to be divided into two domains of modeling.

- Domain Level: where are taking place the necessary activities for solving problems.
- Conversational level: where are taking place the procedures relative to communication, cooperation or division of tasks and other such activities.

The material previously described should be broken down further into two semantic parts. One is the description of the single learners and the other one is the description of a group as a single entity. That division generates two types of models:

- *Individual learner model* that contains learner related information such as knowledge state, motivational traits, learner type.
- *Group models* that could contain the same information as the individual learner model, depending on the definition of a group, but that must additionally contain information pertaining to the complementary and conflicting knowledge, roles of individual within a group or a community and also the state and type of the relationships founds between members.

This model is important because tutoring within a group is far more effective than a one-on-one interaction with a tutor. The behavior of other participant can be used to influence that of a learner. Particular behaviors of other users lead to particular reactions in a learner. Sometime it is very useful to have another user that is acting in a very specific way to help the learner. Yet one cannot predict the behavior of other human users. That leads to the need of an inclusion within ITS of automatic entities that simulate a particular user behavior.

1.1.3. Generalized Pedagogical Agent (GPA)

This pattern's aim is towards the incorporation of intelligent agents in ITS. Those intelligent agents are mainly *pedagogical agents*, autonomous agents that support human learning by interacting with students in the context of interactive learning environments [Johnson et al., 2000]. Pedagogical agents autonomously intervene to gear the state of the learning environment. They can interact with students by providing hints or improving motivation. They can take various roles, such as learner, teacher, co-learner, etc... The GPA pattern (Figure 1.2) is an abstraction and a generalization of those roles.

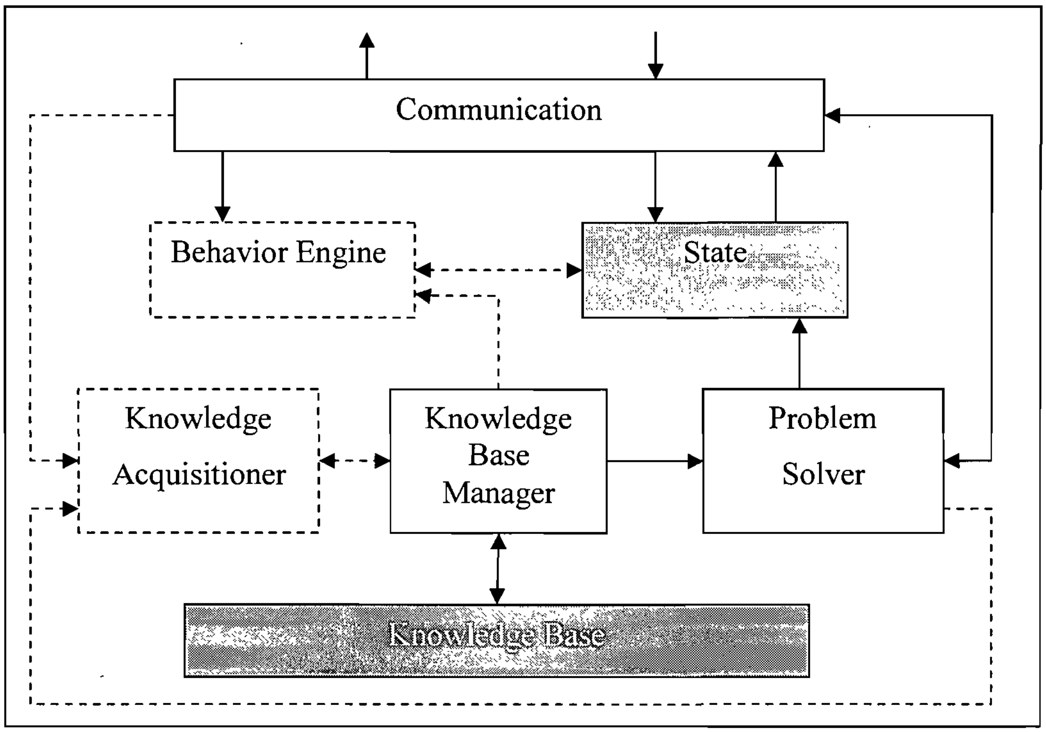


Figure 1.2 GPA pattern (dashed lines show optional elements, shaded boxes components with highly variant specifications) [Devedzic, 2005]

The modules represented in full lines (Figure 1.2) are, at various level of abstraction, necessary for the design of the pedagogical agent.

- The *Knowledge Base* contains the domain knowledge, pedagogical strategies and the student model. It is found in such instances as Learning Tutor agent [Hamburger and Tecuci, 1998], Disciple agents [Tecuci & Keeling, 1999], pedagogical actors [Frasson et al., 1997; Frasson et al., 1996].

- The *Knowledge Base Manager* intervenes when external actors need to use or modify the information stored in the knowledge Base.
- The *Problem Solver* is any kind of engine that directs the agent when facing a problem. This could be a tutoring engine [Hamburger and Tecuci, 1998], or a module that informs the agent when and how to take an action [Frasson et al., 1997].
- The *Communication (Interface)* is a module for perception of the learning environment. This allows the agent to recognize situations where it can intervene.
- *State* is the description of all possible states of the agent. Depending on the type of agent, the state could be emotional [Yin et al., 1998], mental [Paiva and Machado, 1998] or it could be social and describe the agent's relationship with other agents [Vassileva, 1998].
- The *Knowledge Acquisitioner* is responsible for modifying the knowledge of the agent over time. This can be done using diverse machine learning techniques. Case based reasoning [Kolodner, 1993] can also be used to help the agent adapt to new situations.
- The *Behaviour Engine (Expression Engine)* works together with the Communication module and the Knowledge Base to change the agent current state. It could be an Emotion generator [Yin et al., 1998] or a social behavior generator in a multi-agent system [Vassileva, 1998].

Those modules are those that one will be expecting in an agent based ITS.

1.1.4. Co-Learner pattern

One particular derivation of the GPA pattern is the *Co-learner* pattern [Self, 1988]. This pattern includes along with the learner an artificial participant who simulates the behavior of a peer in an ITS. That inclusion has been proved to have a positive impact on learning. It ensures the availability of a collaborator and encourages the student to learn collaboratively, to reflect on and articulate his past actions, and to discuss his future intentions and their consequences [Goodman et al., 1998].

The generic term of "Co-learner" can be applied to various kind of artificial peers, depending on the role they have in the ITS and the type of their interaction with the learner. A Co-learner can be:

- A *learning companion* [Chan and Baskin, 1988] which is learning in the same way as the learner. It generally has the same level of knowledge in order to share peer-like advice when presented with a particular material.
- A *troublemaker* [Aimeur and Frasson, 1996] whose aim is to challenge the student learning by providing solutions that could be right or voluntarily erroneous. This is a way to build up on the student self-confidence.
- Other *reciprocal tutoring* roles [Chan and Chou, 1997]
- An *observer* or a *diagnostician / mediator* [Harrer, 2000]. Those roles are geared towards collaboration and communication inside a group of learner.

A Co-learner pattern focuses on the various interactions and the communication between three types of agents: the *Tutor (T)*, the *Student (S)* and the *Co-learner (C)* agent.

T → S – The Tutor suggests tasks to teach the material. It can provide explanation by giving help, hints, and justification or comments on solutions. The Tutor can base its actions on *Teaching / Pedagogical Strategies*. The Tutor also develops the *Student Model* (this concept will be explained in more details in section 1.2).

S → T – The student requests help as well as provides solutions.

T → C – This interaction can be similar to T → S in the case of a learning companion [Chan and Baskin, 1988]. But a troublemaker agent generally directly accesses the domain knowledge in the same way as the tutor. Otherwise the Tutor builds along with the Student model, the Co-learner model.

C → T Usually similar to S → T but with less importance since focus is not on teaching the Co-learner, but simply maintaining it to a level where its interaction with the student would be pedagogically beneficial to the latter.

S → C – The student can ask for assistance or provide help when requested. That is generally a mean of taking the student through the process of retracing his own steps in order to confirm his learning or discover and fix mistakes in his reasoning.

C → S – This is similar to S → C for a learning companion, but can be different if the Co-learner is a troublemaker. A survey of Co-learners functions and behaviors can be found in [Goodman et al., 1998].

1.1.5. Other patterns and discussion on ITS

All of those patterns as well as some others such as *KnowledgeModel View architecture* (where different views of the same material are presented to the students) are described in more details in [Devedzic, 2005]. Many more patterns can be developed and an exhaustive survey would go beyond the scope of this study. Yet those described here are the main ones found in ITS.

All of those patterns have been developed according to various pedagogical theories. Their aim is to improve learning in an ITS. There is however some debate on the benefit of such systems, with some proponents claiming significant improvements in learning [Woolf and Regian, 2000] but other studies being more critical [Russel, 1998]. From asking questions about the application of multimedia, such as images, sound or videos, the focus has shifted towards asking about the way students learn via the new technology [Clark, 1983]. The aim is to exploit the considerable potential that customized learning would offer [Reigeluth, 1996]. This emphasizes the importance of concentrating on the factors involved in making a pedagogical decision. It follows that the only viable way to make decisions about instructional strategies is to do so dynamically using a system that is constantly observing the student and is capable of continuously updating information about the student's progress, attitude and expectations [Winn, 1989]. This lead to the necessity of focusing on the student model as it is an important part of an ITS.

1.2. Student model

1.2.1. Cognitive Profile

The student model provides information about the student's knowledge and skills. It is used to guide an ITS in making decisions to find the best tutoring approach, so as to give instructions tailored to the best advantage of the student [Ross et al, 1987]. The student model should represent the student's domain knowledge as well as the student's individual characteristics or the cognitive features [Vassileva, 1990]. That is known as the Cognitive Profile. That definition led to the division of the Cognitive profile into the following two sets of information:

- Domain independent information which may include learning goals, cognitive aptitudes, motivational states, preferences, learning styles, and factual and historical data.
- Domain-specific information which represents the student's current state and level of knowledge relating to a particular concept [Brusilovsky, 1994].

Domain-specific information is modeled using several techniques such as overlay models, differential models, perturbation models (Figure 1.3), and episodic learner models.

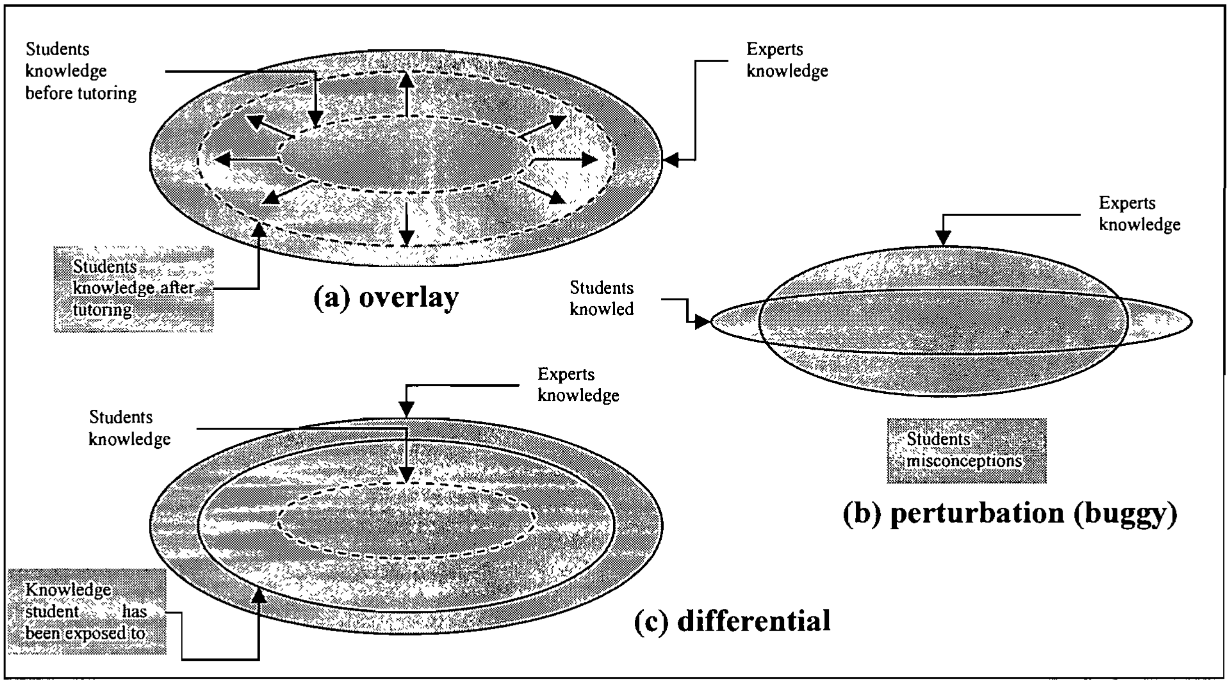


Figure 1.3 Student models: overlay, perturbation and differential

- The overlay model represents the student's knowledge as a subset of the system's knowledge of the subject otherwise known as the domain knowledge. The student knowledge gradually increased and ideally eventually matches the domain knowledge. That is the type of representation used by GUIDON [Clancy, 1983]
- The differential model, in addition to representing the knowledge that the student possesses, keeps track of the knowledge that the student is exposed to. One application that uses that type of representation is WEST [Burton and Brown, 1982]
- The perturbation model can be subdivided in two different parts. The first is identical to the overlay and the differential model. The second is the set of false knowledge or misconceptions held by the student that could lead to erroneous interpretations of the presented material. BUGGY [Burton and Brown, 1978] is an ITS that used that model.

- Another type of student modeling produces the episodic learner model (ELM) [Weber, 1996]. ELM is a type of user or learner model that stores knowledge about the user (learner) in terms of a collection of episodes. In the sense of case-based learning, such episodes can be viewed as cases [Kolodner, 1993]. To construct the learner model, the code produced by a learner is analyzed in terms of the domain knowledge on the one hand and a task description on the other hand. This cognitive diagnosis results in a derivation tree of concepts and rules the learner might have used to solve the problem. These concepts and rules are instantiations of units from the knowledge base. The episodic learner model is made up of these instantiations.

  "In ELM only examples from the course materials are pre-analyzed and the resulting explanation structures are stored in the individual case-based learner model. Elements from the explanation structures are stored with respect to their corresponding concepts from the domain knowledge base, so cases are distributed in terms of instances of concepts. These individual cases--or parts of them--can be used for two different purposes. On the one hand, episodic instances can be used during further analyses as shortcuts if the actual code and plan match corresponding patterns in episodic instances. On the other hand, cases can be used by the analogical component to show up similar examples and problems for reminding purposes."

  The text above is an abstract from: Gerhard Weber and Marcus Specht ' User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems'.

Therefore, the student model is an important component of an ITS since it is via that model that the tutor adapts the training environment in order to answer the needs, objectives and interests of the learner.

1.2.2. Psychological Profile

Over time, the student model has gradually evolved. At the beginning, it merely included the cognitive profile, the learner's knowledge. Later on came up the thought that an adaptive educational system should incorporate pedagogical strategies, and be able to apply different strategies based on a student's psychological profile, [Sternberg, 1997]. Psychological characteristics related to learning, such as motivation, sensory preferences and learner's personality were introduced into the building of student profiles. That extension is also called the psychological profile. Systems developed thus have the ability to offer customized and dynamic teaching. Features of student's psychological profile or learning style are employed as basic elements of customization [Weber et al., 2004].

1.2.3. Emotional Profile

The necessity to take into account more information appeared with recent studies on learners' emotional states. In particular following the argument that putting emotions into machines makes them more human and therefore should improve human-computer communication and lead to a more human decision making process. In 1995 Rosalind Picard proposed a new way to tackle the problem: Affective Computing [Picard, 1995]. She suggested computational models for affect recognition and described new applications of affective computing to areas such as computer assisted learning, perceptual information retrieval, creative arts and entertainment, and human health.

Most current researches in user modeling explore the links between emotions and learning. A lot of these works focus on predicting learners' emotions during their interaction with an ITS. A few systems have been developed bearing that concern in mind. At first very tentative with Conati who used a probabilistic model based on Dynamic Decision Networks to assess the emotional state of the user with educational games [Conati, 2002]. Lately, research has been focused on the benefits of inducing in a learner the emotional states that would optimize learning [Chaffar et al., 2004]. A first multi-agent system was developed that predicts the emotional state of a student based on his brain activity [Heraz et al., 2007].

1.2.4. Cultural Profile

Yet the introduction of emotion in ITS, while taking us one step forward in the elaboration of a picture-perfect student model, brings up new questions of its own. In 2004 [Scollon et al., 2004] write that there is some variability in the frequency to which individuals are expected to feel some emotions depending on the culture of those individuals. The variability is more apparent for positive emotions than negative. Furthermore, the classification of an emotion as positive or negative also depends on culture [Kim-Prieto et al., 2004]. This introduces the importance of culture when trying to model a student. How does that affect e-learning?

Not only emotions, but also many other components of a student model, such as the psychological profile can vary tremendously depending on the culture of the learner. Culture or a student's cultural background is a component that should not be disregarded. Research even show that **the lack of cultural adaptation is a leading reason why e-learning fails to work for a globally distributed audience** [Dunn and Marinetti, 2004]. Therefore, it is necessary to add another variable to the student model: a cultural variable, one that works at a higher level than other components.

One of the first suggestions of such a component was given by the *Culturally AWAre System* (CAWAS) [Blanchard and Frasson, 2005]. CAWAS is a system that generates a cultural profile on the basis of the Hofstede's system of values [Hofstede, 2001] which represents national cultures with a set of dimensions and associated scores. Those dimensions are Power Distance Index (PDI), Uncertainty Avoidance Index (UAI), Individualism (IDV), Masculinity (MAS), and Long Term Orientation (LTO) and scores for those dimensions have been obtained for 50 different countries. It is a rule-based system that is open to taking more rules in order to refine the cultural profile that it generates. To each rule generated is applied a weight that corresponds to the relevance of the rule. Such a system, while being a first answer to the problem at hand, faces two problems.

- First is the validity of the rules. As any rule-based inference engine, this system is dependant on the fact that those rules based on national cultures be trustworthy. There is an infinity of possible "cultural rules" that could describe a national culture. How can one decide which ones are really meaningful and which ones are not? Not every rule shows with the same relevancy or applies with the same weight. How does one come up with a truly reflective representation of those weights?
- Second, is the restriction of that system to the notion of "national culture". As we will explain in section 2.2.2, the definition of an individual's culture extends beyond the scope of his nationality. There is a professional culture for individuals who share the same occupation. There is a social culture that incorporates individuals of the same social class and who tend to share the same activities. "*The trinity of Territory-Culture-Identity is slowly discarded for a more contemporary view of culture. Indeed the former distinctions solely based on geographical distance are growing obsolete, leaving way to less tangible characteristics such as social class and level of education or level of mobility*" [Urry, 2002]. CAWAS reveals very restrictive when dealing with many aspect of culture. An exhaustive cultural profile should take into account those subtler variables.

One should find a solution to the problem of cultural profile that does not face those two inadequacies. The problem is not simply building the cultural profile, but, building it in a way that it includes information that is not only national, but also social or professional. Yet, can one realistically include every single one of the elements that potentially make up a learner's culture in a rule-based engine? Rule-based engines are mostly efficient in building expert systems that emulate the methodologies followed by human beings when solving problems. The heuristics or algorithms they implement attempt to match the reasoning pattern of humans in order to follow the determined steps or repetitive pattern of a causal model. Such methods are known as rule-based, and prove extremely efficient in a domain where a causal model can be established. Finally they come short of a solution in domains where a causal model cannot be determinatively established. In other words, if a causal model is well known the system will probably come up with at least a very close to optimal solution, but if the steps of reasoning are not clearly known, as they are in the case of determining all of the elements that make up a culture, then such a system is hardly applicable. It cannot make use of the human characteristic of intuition when Cartesian reasoning fails to provide satisfying results. In other words, rule-based engine do not seem applicable in determining an individual's culture because there are no known rules, no known steps for solving that problem.

One must turn toward a philosophy that would allow ignoring the knowledge specific to each domain of application and rather than trying to build a model for the path from a problem to a solution, simply link these two. Rather than trying to determine the rules to follow to discriminate between cultures, could one find a way to simply link an individual to his culture? The role of the human expert would be to explain the link, that is, if it is first necessary and even also possible.

There are many problems where there is no causal model directly present to act upon. Some of these domains are those where the level of subjectivity is so high that it does not allow the generation of general rules. An instance is that of taste. Many systems that try to guess the taste of customers, whether about news or entertainment, face that issue. They are named recommender systems. Could one apply those techniques to the task of trying to "guess" an individual's culture? In order to respond to that question one must first take a look at the techniques used in those systems.

1.3. Recommender Systems

Recommender systems try to predict the preferences of customers, whether about news or entertainment. It has now become a consensus that this intuition can only be achieved using some specific techniques. Those techniques allow guessing a system user's preferences. This section discusses those techniques through a quick overview of content-base filtering and particular focus on collaborative filtering.

1.3.1. Content-based Filtering

In the past decade, the "web" has grown to become a major source of knowledge. Yet that knowledge is present in the form of raw data, which could be textual, pictorial or any other form of multimedia within which information is hidden. The increase of the number of users causes an exponential growth of that information which in turn causes a proportional increase of noisy data. It has become more and more tedious to sort out the irrelevancies and retrieve the appropriate information. Following the development of the Internet, many tools for browsing through the data and searching for specific information have emerged. Such tools as Yahoo, or Google in its first versions are very popular for querying web pages. Yet their effectiveness is disputable for they do not necessarily return documents that are relevant to the user, or classified in anyway according to his preoccupations. Those search engines rather make classifications based on popularity, which is a feature that is only related to the document and can turn into a huge bias, as future users will tend to view first those that are ranked first. As the loop repeats itself, information that might be relevant to a particular user, while it is not considered popular by the search engine might become difficult to access because it would have been ranked too low. As a response to that concern, the IT community has created systems that take into account the profile of a user when surfing the web and especially classifying information as relevant or not to the query. Such systems are known as recommender systems. Two main categories of recommender systems can be distinguished based upon the strategies they use to make their recommendations. Some recommend pages based on a social profile of the user, using a technique called collaborative filtering. They will be discussed in section 1.3.2. The others base their recommendations on the very nature of the object being recommended. In most cases, the textual content of documents, but as opposed to simple search engines the relevance of those documents is strongly connected to the user's profile. The technique used is content-based filtering. More precisely, it consists in analyzing the content of information sources that have been rated to create a profile of the user's interests in terms of regularities in the content of the information that was rated highly. This profile may be used to rate other unseen information sources or to construct a query of a search engine [Pazzani, 1999].

The difficulty with such an approach is threefold. First the system must find a way of describing the profile of a given user. This profile must also be determined dynamically to allow for possibility of change or fine-tuning in the preferences of that user. Secondly the system must be able to describe in a standardized manner any document, regardless of language, length, or structure. Then most importantly, those descriptions have to permit comparisons between the user's profile and a document that is to be referred to him. Generally the user's profile is made up of some keywords related to the documents that he has visited and rated. The next step is to compare this profile to the document not yet visited, information retrieval provides with three distinct techniques for meeting up with that challenge. One can either create a Boolean model, a vector space model or a probabilistic model or also lately data mining results for the treatment of documents.

The Boolean model is based on the set theory. The user's profile is modeled as a conjunction of the keywords extracted in a way or another from the ratings he has made in the past. For instance, information consistently found in the documents rated negatively is to be sorted out. The profile thus created is then used as a query made of these keyword and Boolean operators. A query described as <Item1> AND <Item2> would return data containing both these items, while <Item1> NOT <Item2> would sort out data containing either both items or just <item2>. The problem with the Boolean model is that it returns exact matches for the query. It is considered good for information filtering, for instance filtering content only suitable for a mature audience, but would fail at retrieving knowledge [Baeza-Yates and Ribeiro-Neto, 1999].

The vector space model [Salton and McGill, 1983] consists on the representation of the user profile as a vector of terms built from the document they have visited and their ratings of it. The same is done for the documents that are to be evaluated. Usually those documents are fetched in advance and treated before the user requests it. In order to allow comparison, those vectors are of a definite size, regardless of the length of the document they represent. The choice of the set of words is therefore to be determined by the system based on different heuristics. A weight is associated with each term of the vectors. This is usually done using the term frequency–inverse document frequency method or tf-idf (cf. Appendix C.1.) on the occurrence of the terms in the documents. The cosine method is used to compare the angle between the user's model vector and the document vector. The ranking is done on a scale of zero to one: one indicating exact similarity, zero meaning that no term could be matched. This is more interesting than the Boolean method because partial matching can be performed.

The third method is the probabilistic model, and is based on the probability ranking principle. A great set of variables is used to estimate the probability of the relevance of the document to the user's query. Those documents can then be ranked in order of decreasing relevance [Fuhr and Buckley, 1990].

Many other applications of content-based filtering have been developed, in domains as various as news filtering with NewsWeeder [Lang, 1995] or book recommendation LIBRA [Mooney and Roy, 2000] that collects from Amazon meta-information such as title, authors, synopses, published reviews, customer comments, related authors, related titles, and subject terms on books, rather than actual textual content. Yet they all suffer from the same problems.

One problem of major importance is that it does not permit to discover new themes. The system keeps recommending items of the types that the user has just seen. There is a risk of overspecialization and also of isolating the user from information he might be interested in, but has not seen yet.

Also if a profile is generated from using the system, and is refined with experience, there seems to be a problem for recommending a first time user, strictly based on content-based methods, because in that case, the profile is blank.

Another point is that the basis for recommendation is simply the textual content of the items rated. The system implies that if the user likes a certain web page for instance, it is because of the words it contains. It does not take into consideration the structure, style or point of view of the source of information rated [Shardanand and Maes, 1995].

Finally the main drawback of that technique is that it does only permit to rate either textual item, or textual descriptions of other types of items, because it is based on counting a frequency of words. Yet the web has grown into more than that. Information is now present in various multimedia forms. In those cases, content-base filtering is very dependant on the accuracy of the textual description of those objects.

A next alternative that has risen in popularity in response to those cons is collaborative filtering.

1.3.2. Collaborative Filtering

The first kind of this new type of recommender system is Tapestry [Goldberg et al., 1992]. They introduced the idea that in order for recommendation to be effective, it required the intervention of humans. The system is an email filter. It allowed users, upon reception of any kind of electronic document, to associate annotation to it. Three distinct types of collaborative filtering can be defined: pull-active, push-active and automated.

Pull active collaborative filtering as illustrated in Tapestry [Goldberg et al., 1992] indicates that the user takes it on himself to request the information that he assumes would be interesting for him. For instance a given user could request to see all the emails that one of his coworkers found interesting. The user pulls the information toward him.

Push active collaborative filtering describes the opposite behavior. One popular system [Maltz and Ehrlich, 1995] implements that vision. Their application lets users indicate information that they believe their coworker would find interesting, in the form of a hyperlink. The users are asked to push the information toward the right person. Both active collaborative methods can only work for a small group of coworkers or friends who already know each other and are aware of each other's centers or interest. For a network the size of the World Wide Web, it would be impossible to implement. People across the world could share interest. They would not be aware of it.

The solution is that it is the system itself that keeps track of the users' interest. Such applications use an automated collaborative filtering. GroupLens [Resnick et al., 1994] was the first of that kind of application. It is a net news filter that requires each user to rate the articles that they read. The system then creates a profile for those users based on their ratings of articles. In order to make recommendations for a user A on a news article, the system interrogates users that have had the same tendencies in the past as A about other articles. Following their opinion on the news article, it will be recommended or not. Quickly following GroupLens, a plethora of applications have emerged, based on the same principle. Some of those are Ringo [Shardanand and Maes, 1995] ancestor of FireFly that recommends music or Recommender [Hill et al., 1995] that rate movies through email.

The general framework of all collaborative filtering systems can be broken down to three steps [Herlocker *et al.*, 1999]. The first is to compare the users to all other profiles held in the database; then to select the users that are suited for use for recommendation based on their similarity with the user; finally to calculate a weighted sum of the selected user and make recommendations based on the result. Over the years, in order to increase efficiency, many models and architectures have been developed. They can be grouped in three main categories. *Memory-based*, *model-based*, and lately an interest has been shown toward *peer-to-peer* models.

A memory-based architecture retains the profile of all users in the database and systematically compares a given user to all of them for every recommendation. Different means have been used to achieve that comparison.

The first was introduced by Grouplens, and used the Pearson correlation to make its predictions. It is computed as such:

$$\omega_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \overline{r_a}) \times (r_{u,i} - \overline{r_u})}{\sigma_a \times \sigma_u} \quad (1)$$

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{n}(r_{u,i} - \overline{r_u}) \times \omega_{a,u}}{\sum_{u=1}^{n}\omega_{a,u}} \quad (2)$$

where $p_{a,i}$ represents the prediction for the active user a for item i. n is the number of neighbors and $\omega_{a,u}$ is the similarity weight between the active user and neighbor u as defined by the Pearson correlation coefficient [Herlocker et al., 1999]. $r_{a,i}$ represents ratings for the active user a for item i. The similarity value varies between -1 indicating a negative correlation and 1 indicating a perfect correlation. A similarity of 0 means there is no correlation between the users.

Ringo finds more accuracy in limiting the number of neighbors to those only whose correlation was greater than a certain threshold.

Other than the Pearson correlation coefficient, a vector space model can be used to assess the similarity of users. The principle is the same as that widely used for content filtering. Simply, documents become users, words become titles and frequencies become votes. The vector space model was found less efficient than the Pearson correlation [Breese et al., 1998].

Ringo also used the mean squared differences algorithm to compute the degree of dissimilarity between two user profiles [Shardanand and Maes, 1995].

In every instance of the memory-based model, the system compares the user to all other users. The problem is that there is generally a huge set of items and each user possesses only an infinitesimal portion of that set. The matrix that associates users with items is therefore usually very sparse. This causes the accuracy of the recommendation to be very poor.

As opposed the memory-based, the model-based architecture does not compare a user to everyone else. But rather is deducts a model of the users in the system by various means as various as neural network classifiers [Billsus and Pazzani, 1998], induction rule learning [Basu et al., 1998], linear classifiers [Zhang and Iyengar, 2002], retraining Bayesian networks [Breese et al., 1998], dependency networks [Heckerman et al., 2000], latent class models [Hofmann and Puzicha, 1999], mixture models [Lee, 2001], item-based CF [Sarwar et al., 2001], principle component analysis-based CF [Goldberg et al., 2001] or association rule mining [Lin et al., 2002]. That model is then used to predict ratings. This causes an economy of resources and time for the entire matrix is not worked on every time. The reverse side of the coin is that with every new user or item the model is to be retrained, which is costly.

Lately, a new approach is being taken into consideration. A peer-to-peer approach, according to which there should be no centralized database of users. Each user should hold a portion of the whole dataset. The principle is similar to those used in the peer-to-peer file sharing networks such as Gnutella. In a centralized network, a system administrator can have access to all the user profile. This becomes trickier in a peer-to-peer network, for the source of the information is harder to locate. One of the systems proposed for the support of products and service recommendation for mobile customers [Tveit, 2001], uses cell phone agents as peers and sends users vectors to direct neighbors. This direct neighbor holds a cache model of its neighbors with which it compares the query. If similarity is found, then a response with a recommendation is sent back to the source of the query. If a user is on a path of two similar users, it is suggested that they connect directly. If no similarity is found, then the query is sent forward. The advantage is that each user holds a small amount of the whole information. This reduces computation. There are a few other applications based on the same idea such as a peer-to-peer recommendation for web addresses that hold specified information [Joshi, 2000], or PVT [Cotter and Smyth, 2000] that recommends TV programs.

While providing many solutions, collaborative filtering creates new problems of its own.

Most applications are memory based and require huge matrices for computation. Those matrices are usually sparse with, for instance, a density of 0.03 for the EachMovie dataset [Canny, 2002]. Model-based architectures that reduce the size of the matrix require costly computation for every new item.

Also a problem arises when a new item is inserted in the system. There is no rating yet, therefore it cannot be recommended. In order to go around some of these problems, more and more systems created are some combinations of content-based and collaborative filtering: hybrid systems.

Hybrid systems, as their name suggest are a combination of two or more of the methods seen above. Different ways of combining exist [Burke, 2000], depending on the need. Below (

Table 1.1) are a list of the methods and the problems they respond to, [from Burke, 2000].

Table 1.1 Hybridization Methods [adapted from Burke, 2000]

Hybridization method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation
Switching	The system switches between recommendation techniques depending on the current situation
Mixed	Recommendations from several different recommenders are presented at the same time
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendation given by another
Feature augmentation	Output from one technique is used as an input feature to another
Meta-level	The model learned by one recommender is used as input to another.
Pyramid CF	The system filters successively on data content, then user similarities and finally user credibility

Weighted systems make recommendations with many different techniques and then combine them, sometimes firstly simply linearly [Claypool et al., 1999] or each result from a recommendation method can be considered a vote and then combined [Pazzani, 1999].

Switching systems such as DailyLearner [Billsus and Pazzani, 2000] go back and forth between techniques depending on the situation.

Mixed systems like PVT [Cotter and Smyth, 2000] or ProfBuilder [Wasfi, 1999] and PickAFlick [Burke et al. 1997; Burke, 2000], make recommendations with many techniques and simply present them all to the user. This is possible when rating is not needed.

Feature Combination such as Ripper [Basu et al., 1998] use content-based techniques on data acquired through collaborative techniques.

Cascade is an incremental method that consists in using two techniques. One usually helps to refine the other one. EntreeC [Burke, 2000] is one such example.

Feature Augmentation uses the results of one technique as features for the next technique. Libra [Mooney and Roy 1999] is an example.

Meta-Level means that the whole model generated by one technique, rather than simply the result, is used as input for the next technique. Systems such as those of Fab [Balabanovic 1997, 1998], and LaboUr [Schwab et al., 2001] follow that trend.

Pyramid Content Filtering systems [Razek *et al.*, 2004] successively filter on content relevance then user similarity and finally user credibility in order to make a trustworthy recommendation.

Collaborative filtering is the solution to the problem of forming groups of users that are not based on tacit rules. Those groups are rather composed based on the behavior of the users and their interaction with the system. It is the direct solution to problems that could not be solved with the use of rule-based expert systems. As shown in section 1.2.4 building a cultural profile based on rules faces many problems. Using collaborative filtering would allow overcoming the problem of finding explicit rules for composing cultural groups.

Chapter 2: Suggested Solution

2.1. Problem

"One of the main consequences of globalization in the domain of information is that individuals all over the globe have access to the same global media. However, it is perceived differently depending on the local culture." [Appadurai, 1996]

The issue we are trying to tackle falls within the frame of that general assertion. As far as teaching with ITS is concerned, the problem is more specific and is mainly restricted to e-learning. Indeed e-learning suggests that a huge pool of extremely diverse students has, or at least potentially has, access to the same information. That same material when presented to different learners can create different sets of beliefs, and therefore a completely different knowledge than that expected **if the cultural background of the learner is not taken in consideration when supplying the material**. Answering that concern is the main motivation of this study.

In other words, the role of an ITS is generally to transmit a particular knowledge. Simply presenting the information in a standard manner could lead to misunderstandings and cause students to reach different conclusions even if the information base is the same. For instance, when the Open University of Israel offered courses to 17,000 students in Russia, there were not aware until their study was conducted, that a major reason for learners dropping out of the courses was "cross-cultural misinterpretation of teaching material" arising from "substantial and semantic disparities." [Victoria *et al,* 1999]. This discovery led the researchers to question the cultural assumptions of course developers when offering courses to students in other cultures [Victoria *et al*, 1999]. Pedagogical strategies are a key feature in efficient teaching. Yet pedagogical strategies vary widely across the globe. Therefore for ITS to be more efficient, they need, just like human teachers, to adapt their teaching to the students' cultural background.

2.2. Supporting Theories

2.2.1. Responsive Teaching

"The academic achievement of ethnically diverse students will improve when they are taught through their own cultural and experiential filters" [Au & Kawakami, 1994; Gay, 2000]

That statement borrowed from cross-cultural pedagogical studies is the **main supportive basis of our response to the problem**. The solution aimed at is to build a setting that will appropriately emulate a learner's "cultural and experiential filter". We will tend to focus more on the "cultural" aspect of the so-called filter.

We intend to teach a learner in a manner that will be perceived as "familiar" to the learner, a manner that fits right inside his cultural expectations. Responsive teaching encourages taking the learner through a way that he recognizes, one that gives that feeling of "being home".

2.2.2. Definitions of Culture

An obvious problem that immediately emerges is the following: what exactly creates that feeling of "being home"? What variables are supposed to be changed in order to adapt a teaching experience culturally? Exactly, what is culture?
Literature has given us two leading definitions of cultures:

The first is a "relatively stable system of shared meanings, a repository of meaningful symbols, which provides structure to experience". [Kashima, 2000] In that first definition, culture is seen as a static set of symbols that are interpreted similarly by a group of individuals.

The second definition views culture as "a process of production and reproduction of meanings in particular actor's' concrete practices (or actions or activities) in particular contexts in time and space" [Kashima, 2000]. That last definition offers a view of culture that is more dynamic. It touches ways in which people would act or respond in given contexts.

In this study, the term culture will encompass both those definitions. Meaning it is *both* a static set of symbols similarly interpreted *and* the way in which people react when put in a given situation, a given context.

Therefore if we are to adapt to a particular culture, we need to first understand the manner in which an individual's interpretation of a particular symbol differs from that of members of a different culture. Secondly, we must employ that knowledge to induce the appropriate response (mental or intellectual) that will favor learning for an individual of a particular culture.

2.3. Practical Approaches: C.A.M.E.L.E.O.

The main difficulty that we face is that there are no formulated rules to determine a learner's culture. There are only implicit principles, hard to define. Also, one could argue that the aim of a tutoring system is simply to provide a user with the most efficient teaching possible. Our systems should therefore not necessarily attempt to find rules to deduce a user's culture. Teachers in the real world do not have any rule of thumb for determining their student's culture. They rather classify students into groups. Students from a certain group, a certain cultural background, respond better to a certain type of teaching. Students who have responded similarly in the past are simply put in the same group. The more their responses are similar the "closer" they are assumed to be. A strategy that was productive on a particular student will be assumed to work on a peer that is "close" to him. **Those groups of close peer will be considered group of people sharing the same cultural background**.

This means that a system must recommend the answer provided by the user that has the closest cultural background to the learner's. A clearer definition of the elaboration of a "cultural background" will be provided in section 2.3.1. Another way to put it is that the best expert of a student's background should be the peer that is closest to that student. That peer is the best able to supply advice to the ITS based on his knowledge of the learner's expectations. The peer shares his experience on the application of a strategy. It can be thought of as a *learning companion* [Chan and Baskin, 1988].

This is as close as possible an interpretation of the first definition of culture. Culture is the fact of sharing the same background, and understanding things the same way.

2.3.1. Collaborative Filtering

However, how do we know which individual most closely resembles the learner? Here we need to turn towards the second definition of culture. An individual can be said to belong to the same culture as a learner if that individual has been proved to respond in a similar manner when presented with the same situation as the learner. A measure of that similarity can be obtained through an appropriate use of collaborative filtering.

Based on Notions and Resources associations

The purpose of an ITS is generally to teach a course. A course consists of notions that will be given to the learner to assimilate. Those notions can be represented by a whole range of resources. Those resources can affect all of the learner's senses since ITS currently make use of multimedia such as sounds and images. But more abstractly, a resource could come in any type or shape. We deal with the abstract concepts of a *Resource* that is used to transmit a *Notion* to the user. Notion/Resource parallels in meaning to the Symbol/Meaning concepts discussed in the first definition of culture given in Chapter 2.2.2.

One way to effectively assess the accuracy of a Resource at describing a Notion is to directly ask the learner. The learner should be able to indicate the extent to which a Resource represents a Notion by ranking Notion/Resource associations. Similarity of cultural background can therefore be determined by using collaborative filtering to compare the rank the symbol/meaning association or Notions/Resources associations of all learners. This means that we call "culture" of a learner the set of ranks he has given to the Notion/Resource associations encountered in the course of use of the system. Indeed that set of rank is analogous to a "value system". This is related to the definitions of [kashima, 2000] of a common culture, or common cultural background, implying "shared meaning". Individuals for which share the association of Notion/Resource would belong to the same culture. In a practical way, that set of rank can be viewed as a vector of ranks: the Notion Resource Vector (NRV). The problem at hand becomes the determination of the Resource that best describes a Notion for a given learner. The solution is to suggest the preferred Resource of the user that is most similar to the learner. is an illustration of the way that collaborative filtering can be used with NRV similarity between learners.

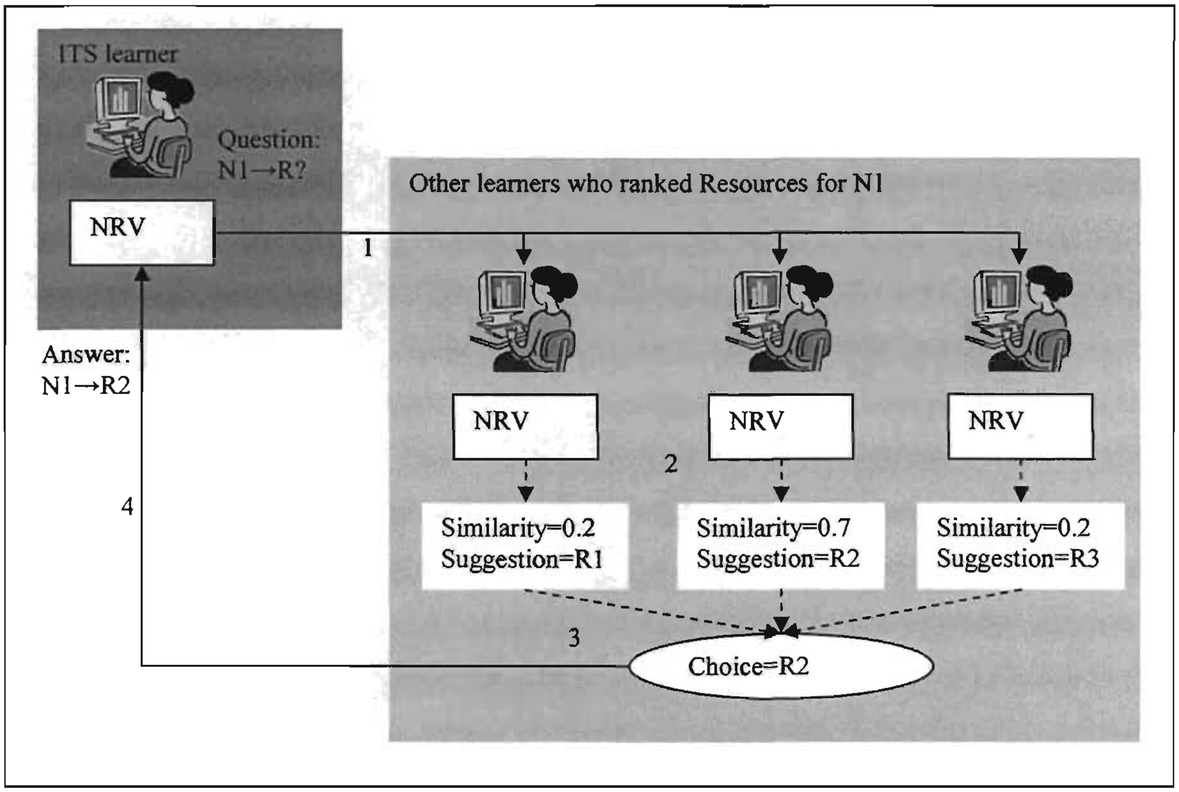


Figure 2.1 Collaborative filtering using NRV similarities

- In this example, in step 1 the ITS learner wishes to find the appropriate Resource for Notion N1. His NRV is sent to the system's other learners that have ranked resources for N1.
- In step 2, each learner computes a value of similarity between their own NRV and that of the learner they have just received. This is done using the Pearson correlation equation. The value is computed using equation (1).

$$\omega_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \overline{r_a}) \times (r_{u,i} - \overline{r_u})}{\sigma_a \times \sigma_u} \tag{1}$$

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{n}(r_{u,i} - \overline{r_u}) \times \omega_{a,u}}{\sum_{u=1}^{n}\omega_{a,u}} \tag{2}$$

Where $p_{a,i}$ represents the prediction for the active user a for item i. n is the number of neighbors and $\omega_{a,u}$ is the similarity weight between the active user and neighbor u as defined by the Pearson correlation coefficient ω [Herlocker et al., 1999]. $r_{u,i}$ represents the rating of user u for item i.

- In step 3, the system selects the resource suggested by that learner with the highest similarity value. In this example it is R2, which is suggested by the user with similarity 0.7. The suggested resource for N1 would therefore be R2.
- Finally, in step 4, the suggested choice is sent back the requester.

We need to specify that in our case, equation (2) is not used. Rather than predicting a rank, the system chooses the resource ranked best for N1 by the user with the most similar NRV.

Based on country of origin Rule Weigh Vectors

The methodology previously described has the following limitation: if a learner has never ranked a Notion/Resource association it is not possible to compute his similarity with the other users.

Collaborative filtering system must use some ways to overcome that limitation. Generally, those systems use some ways to collect basic information on the learner that has nothing to do with their interaction with the system. In our case we must find a way to differentiate users culturally. Therefore a more static definition of culture is needed.

The system can overcome that problem by using a similar system known as CAWAS' implementation of a static culture [Razaki, Blanchard et al., 2006] meaning using the Hofstede's system of values [Hofstede, 2001] which represent national cultures with a set of dimensions and associated scores. Those dimensions are *Power Distance Index* (PDI) that is the extent to which the less powerful members of organizations and institutions (like the family) accept and expect that power is distributed unequally, *Uncertainty Avoidance Index* (UAI) that deals with a society's tolerance for uncertainty and ambiguity, *Individualism* (IDV) that is the degree to which individual are integrated into groups, *Masculinity* (MAS) that refers to the distribution of roles between the genders, and *Long Term Orientation* (LTO) that compares thrift and perseverance to respect for traditions, fulfilling social obligations and protecting one's face. Those dimensions and scores for those dimensions have been obtained for 50 different countries. In an effort of abstraction, those dimensions or other factors that can characterize a group are called Cultural Facts. In CAWAS the scores for those Cultural Facts are stored in a data structure called Rule Weight Vector (RWV) [Blanchard *et al*, 2005]. Cultural similarity can therefore be determined by comparing the values of those dimensions in a manner identical to that described in the previous section. In this case those dimensions stand in place of the Notion/Resource association. The preferred Resource of the most similar user is suggested to the learner.

RWV Methodology

The methodology using RWV is organized around the concept of Cultural Groups. In order to initialize those groups, we can use different cross-cultural works such as

[Hofstede, 2001]. However, those studies are not complete and do not necessary fit with what we know concerning our learners and with what we think is relevant for cross-cultural adaptation. For instance information relative to some groups might be missing from those studies. Secondly we might learn additional information about ways to differentiate groups. Therefore we have built a tool for creating a new kind of Cultural Group (Figure 2.2). This tool called Cultural Inheritance Manager (CIM) allows us to create new groups from scratch by attributing scores to cultural facts.

Group Inheritance (adapted from [Razaki et al, 2006].)

We also introduce a mechanism of group inheritance, which means that some groups are specialized versions of broader ones. For instance one can sense that the population from Martinique, although being classified as French if we follow the political subdivision suggested by Hofstede, would share some important values with groups from other parts of the world (such as Africa), that are not found in the general French population. When creating a group that inherits from multiple other groups, we affect Inheritance Weights to each parent, which are weights that describe the level to which the child group relates to his parent groups. Needless to say that it takes an expert to define those weights. A child group is constructed in the following manner:

- For such a group the scores of the Cultural Facts are automatically inherited from its parents.
- In case of conflict, a situation where two or more parents present the same Cultural Fact with different weights, it is the system's administrator's task to break ties. They can either chose the Cultural Fact coming from one parent and discard the others or compute a mean of those Cultural Facts scores, weighed by the Inheritance Weight corresponding to each parent, and associate the obtained result to the corresponding Cultural Fact in the child group.

- Additionally, specific Cultural Facts can be added to a group, which means that those new facts cannot be found in any of the parent groups.

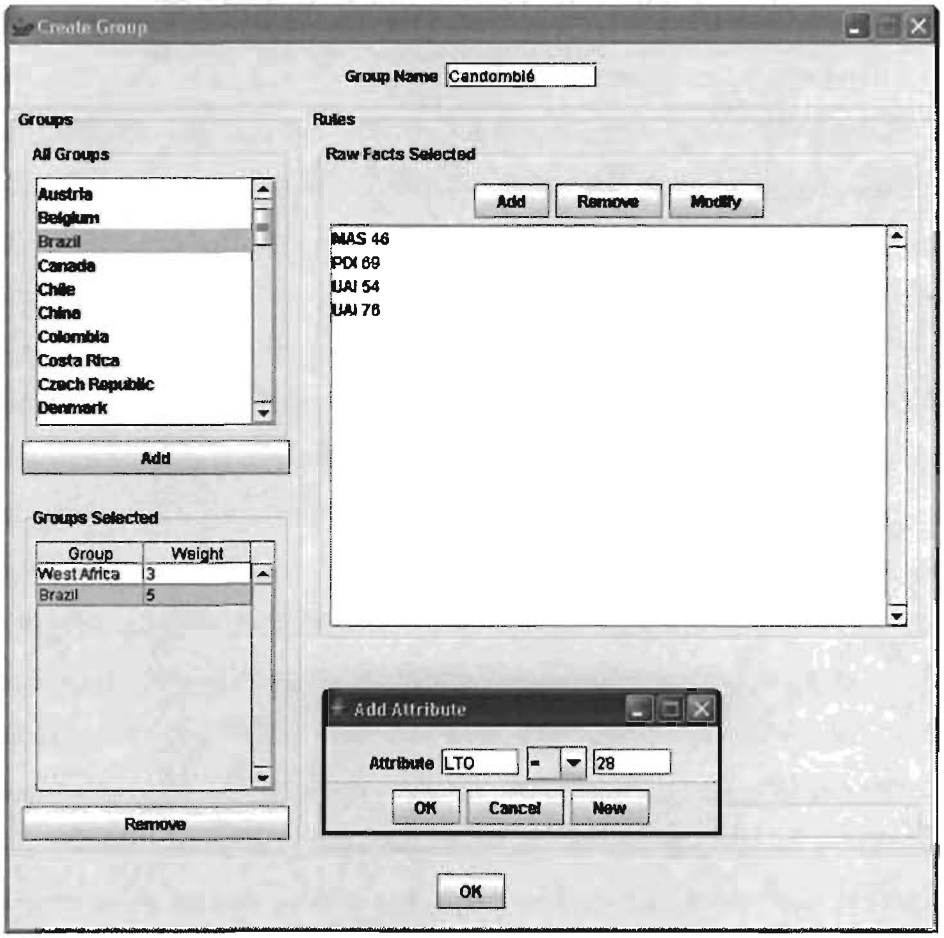


Figure 2.2 CIM: a tool for the creation of Cultural Groups

The group "Candomblé" displayed in Figure 2.2 is defined by Wikipedia as "*an Afro-American religion practiced chiefly in Brazil but also in adjacent countries. The religion came from Africa to Brazil, carried by African priests and adherents who were brought as slaves between 1549 and 1888*". It comes to no surprise that adepts of that religion display a culture that is a hybrid of African and Brazilian cultures.

The tool permits to create this new group from those already present in the system and determine its Inheritance Weights. In this case, the Inheritance Weight of the "Candomblé" group to West Africa is set to 3 by the system administrator and it is set to 5 for its relation with the "Brazil" group. Table 2.1 presents Cultural Facts scores for the groups Brazil and West Africa obtained from [Hofstede, 2001]:

Table 2.1 Cultural Facts scores for Brazil and West Africa

Group	PDI	IDV	MAS	UAI	LTO
Brazil	69	38	49	76	N/A
West Africa	77	20	46	54	N/A

As we can see with Figure 2.2 and Table 2.1, in order to initialize the new group, the user has chosen to select the score of PDI from Brazil but the score of MAS from West Africa. The score of his Uncertainty Avoidance Index (UAI) has not yet been determined (it is still displayed twice in the interface in Figure 2.2). The system will require the user to specify that score before validating the creation of the group.

Three types of operations are possible:

- It is possible to add and set a score for a Cultural Fact that is not present in the parent groups. In the example presented in Figure 2.2, a score for "LT0" is being added to the "Candomblé" group.
- One can also choose to modify the score of a Cultural Fact selected from one group in order to better suit the newly created group.
- Lastly, it is possible to altogether remove a Cultural Fact from the new group, meaning that even though it is present in one of the parents groups, it is not relevant or unknown for the child group. In the example of Figure 2.2, there is no score related to the IDV dimension for the new group although it is present in both its parents because the author does not think that the IDV of the "Candomblé" group is

related to those of its parents and he does not dispose of any information that could help him to initialize the IDV score.

Once those selections have been validated, the new group will be created. The methodology for cultural adaptation is in no way dependant on CIM. The purpose of the CIM is simply to give a little more flexibility to what some might find as an otherwise very rigid definition of cultural groups given by studies such as Hofstede's.

The CIM allows us to have a higher control over the initialization of a student using his RWV. NRV and RWV are the component of the learner's *Cultural Student Model* (CSM). The definition of the CMS is extremely important because it is the component upon which the system will base its choice when comparing users' cultural backgrounds.

Note: The description of CIM is adapted from [Razaki *et al*, 2006].

Random

Yet, even the methodology using RWV fails in one case:

- The case when a learner comes across a Notion/Resource association that has never been ranked by any other learner.

If all fails the system reverts to a random assignment of Resources.

Implicit rating

The steps described up to this moment involve the learner's explicit appreciation of the correspondence between the Notions and Resources that are shown to him. Our system should go further in order to add a variable that takes in consideration whether the Resource chosen helps to improve learning. Indeed, the main role of an ITS is to teach. Resources that must be chosen are those that yield the best possible understanding of the material. If the ITS provides a way of testing the learners knowledge, that variable should be combined together with the rank of a Notion/Resource in order to refine it.

The methodology thus described, up until the "implicit rating", is analogous to that of a *Switching Recommender System* (described in section 1.3.2) for it goes back and forth between different kinds of recommendation techniques. The implicit rating adds a *Feature Augmentation* aspect (described in section 1.3.2), even if it is a minor one, for it uses the output of the recommendation together with another technique to discriminate further between alternatives.

This is our suggested solution to handling the problem of adjoining a cultural adaptation process to the mechanics of an ITS. It is named *C.A.M.E.L.E.O.* (Cultural Adaptation Methodology for E-Leaning Environment Optimization).

Chapter 3: CAMELEO's Architecture

3.1. Integration to an ITS

The theoretical model leads to the design of the following architecture of a web-based expert recommender system, also named *CAMELEO*, that fits within the structure of an ITS as shown in Figure 3.1.

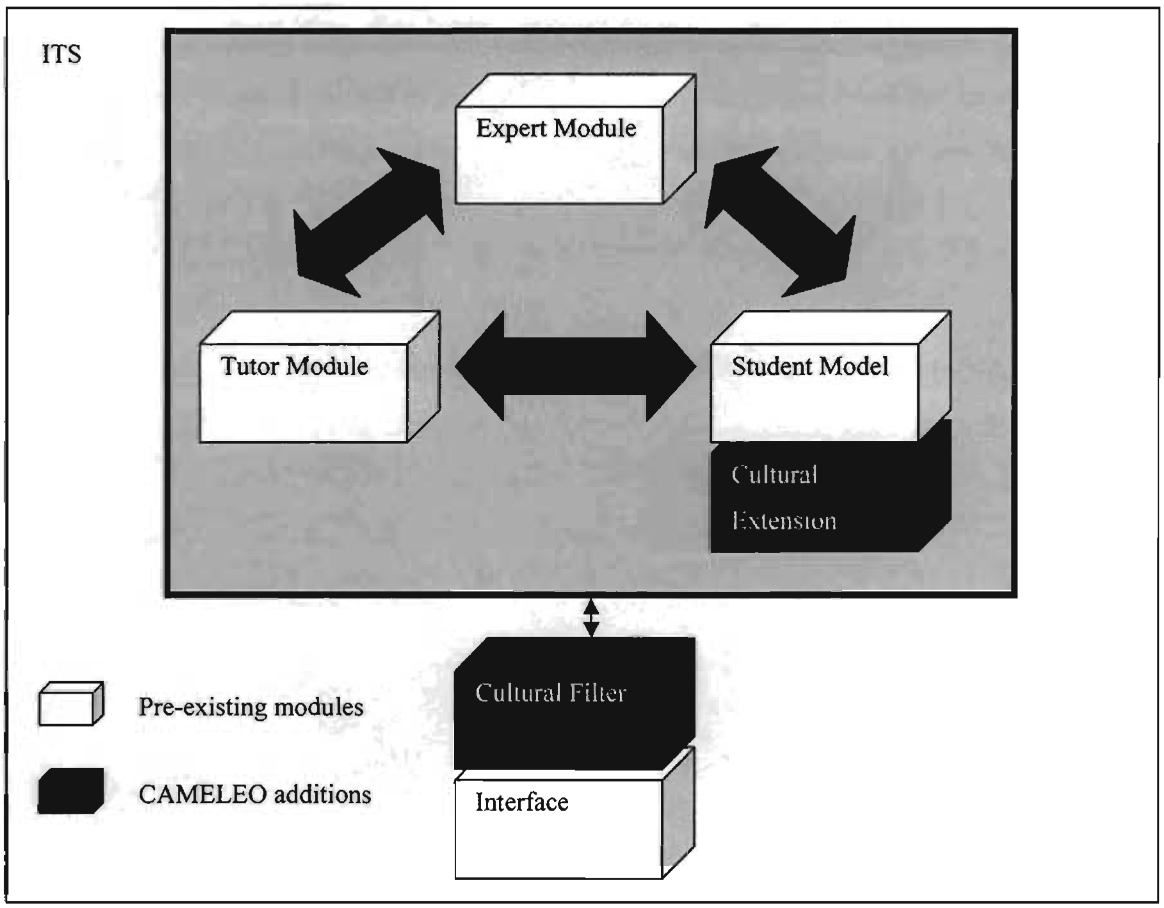


Figure 3.1 – CAMELEO expansion of ITS - derived from [Burns and Capps, 1988]

As shown in Figure 3.1 CAMELEO consists of two parts: an extension to the student model and an additional filter to the interface of an ITS.

Cultural Extension

The cultural extension is a representation of the cultural data that characterizes each learner. It consists of:

- A representation of the Notion/Resource couple, and the rank associated to them, with a data structure holding the set of ranks and called Notion Resource Vector (NRV).
- A representation of the dimensions of Hosftede and their values for each learner according to her place of origin in the form of a vector holding the set of values and known as Rule Weight Vector (RWV) [Blanchard, Razaki et al., 2005].

Cultural Filter

The cultural filter uses the C.A.M.E.L.E.O. (the methodology) on the data present in the cultural extension in order to appropriately adapt the interface to the learner's culture. The remaining of this document describes the details of the cultural filter.

3.2. Cultural Filter

3.2.1. Choices and justifications

3.2.1.1. Three-tier architecture

The aim is to build a system that recommends different resources to its users depending on their background. Therefore its end users are facing different interfaces, while the basic data is the same. There should be a certain level of flexibility of the interface that is independent from the data.

Additionally, the use of collaborative filtering implies that the greater the number of users the more precise the predictions of the system. The system being web oriented, it ought to be ready to welcome an enormous number of users. This also implies that there is a need for the possibility to upgrade and modify data without necessarily affecting the logic of the interface. We are facing issues of flexibility, maintainability and scalability.

Consideration of those facts call for a three-tier architecture. We need the implementation of a middle tier to manage resource allocation and data management.

3.2.1.2. Multi-agent system

The system works using information relative to user's behavior and culture. In this era of high privacy awareness, we must not take lightly the manner in which we deal with the information we gather on our users. We must be able to encapsulate that sensible information within a trustworthy structure and only share the part of that information that can be available to the public. The user should be able to have a structure that works autonomously on his personal information and decides which parts of it to share.

A multi-agent architecture conveniently answers our needs for both autonomy and encapsulation of information.

3.2.1.3. General presentation of modules

As explained earlier, the choice made was that of a three-tier architecture. The modules present in each tier (those will be discussed in much more depth later, cf. section 3.2.2) are grouped as such:

- The data tier contains:

  - The DB: a module that contains all the data access functions
- The client tier contains:
  - The web browser
- The application tier contains:
  - The web server
    - A Template Converter (to be defined in more depth)
    - Proxy
  - The server agent platform
    - Culture Modeler (an agent)
    - Allocator (an agent)
    - Sleepys (agents)
    - Clients (agents)
    - Proxy Agent (an agent)

Inter-module communication is as shown in Figure 3.2.

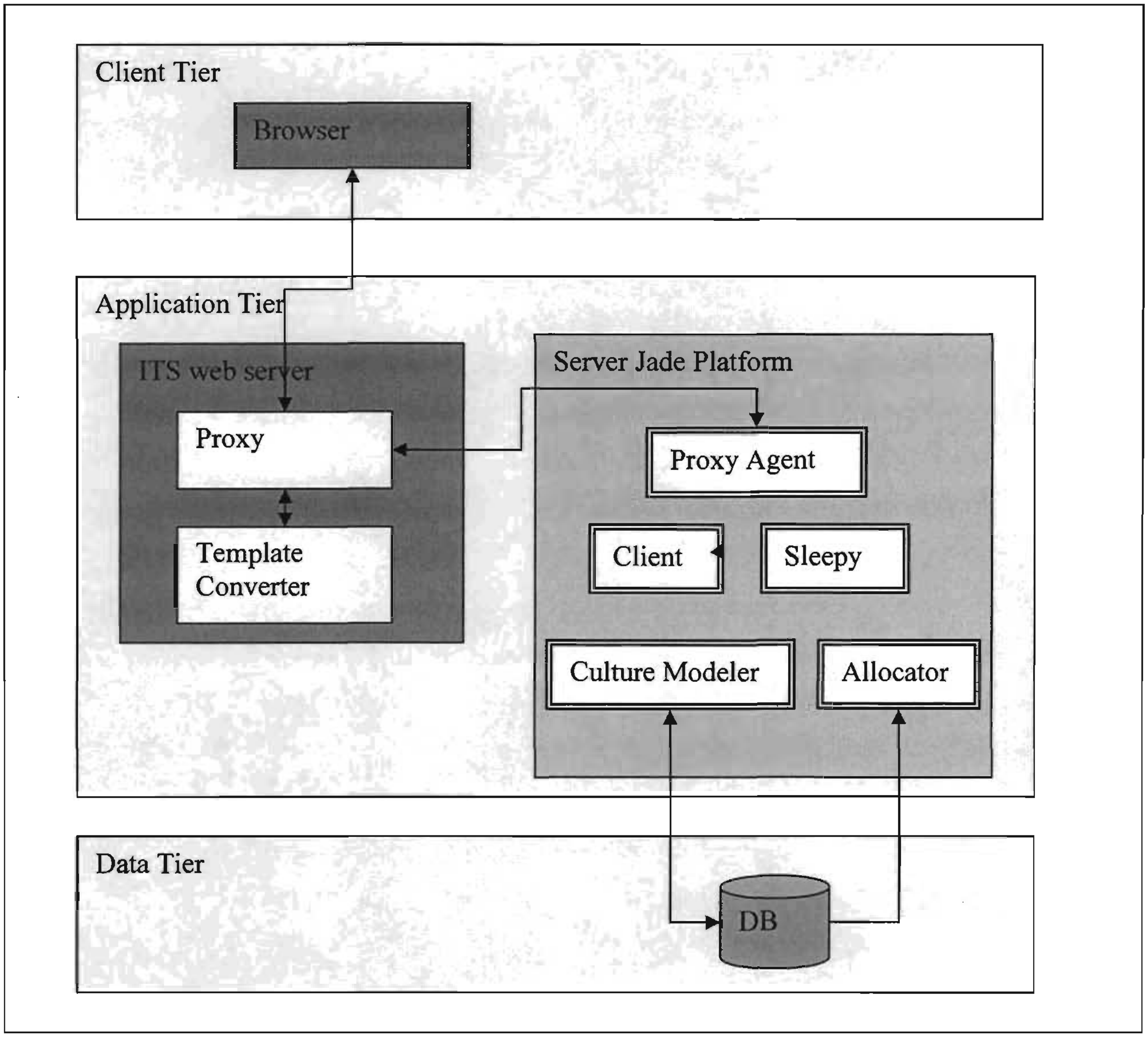


Figure 3.2 Functional Architecture – CAMELEO's Inter-Module communication

Precision need be added that the agents can potentially all communicate with one another.

3.2.2. Modules description

This section gives a description and justifies the logic behind the choice of each module. The exhaustive list of function operated by each module can be found in Appendix A.1.

Browser:

- Description: It is the module via/which the user remotely interacts with the ITS.
- Justification: This allows for a web based system where many users can connect around the world.

Template Converter:

- Description: The module that contains a description of the pages that the ITS will show the user. They are described in a structure called Cultural Templates. The module converts Cultural Template to HTML code. (This will be explained more in detail, cf. section 3.2.3)
- Justification: Allows having one generic template of each page for all users. Those templates will be modified accordingly to values of session variables which in turn depend on the user profile.

Proxy:

- Description: The module simply creates a Proxy Agent that links the web server to a Jade platform. This stated, from now on and for clarity purposes, there will be no distinction between Proxy and Proxy Agent.
- Justification: Allows the use of servlets that permits to communicate with jade agent via a web server.

Allocator:

- Description: Manages Client Agents

- Justification: A centralized manner to control whether an agent is already logged in and to allow secure communication between agent by hiding agent ID when sensitive information is sent.

Client:

- Description: Request the Cultural User Model
- Justification: uses agent to efficiently encapsulate information about the user model.

Sleepy:

- Description: Update list of Agent that have are logged in the system. Sleepy agents keep CSM for all users that are not logged in. For efficiency purposes, each Sleepy can only keep up to ten CSM.
- Justification: Answers the need to keep track of all Client Agent whether or not logged in, for computational purposes. Also the type of operation that can be performed on a Client may differ depending on the fact that he's logged in.

Culture Modeler:

- Description: updates and communicates with the DB
- Justification: all communications with DB are via that agent for synchronization control.

DB:

- Description: Manages the information stored in the database
- Justification: means of information storage.

3.2.3. Template Converter

A particular module that needs special attention is the Template Converter. It works as follows: a basic skeleton is created for each page that would be displayed. The "skeleton" is the template file that is meant to be adapted accordingly to the cultural background of the system user. It is called the Cultural Template (CTP) file.

In other words a Template Converter contains the code that converts a CTP into a script that can be understood by the web browser. The code within a Template Converter is both dependant on the CTP that it is trying to convert and the type of format it must be converted into. The template converter is mainly an interpreter that fills in the gaps left for cultural adaptation and replaces them with the appropriate Resource.

3.3. Main Functionalities

In which way do the modules previously described cooperate in order to put in practice the C.A.M.E.L.E.O (the methodology)? This section gives an overview of the principal functionalities of CAMELEO (the system). The exhaustive list of module functions can be found in Appendix A.1. We will cover five of those functionalities which are:

- Signup
- Login
- Solve Page
- Rank Resource
- Evaluate Lesson

3.3.1. Signup

Process description:

Signup is the process from which the learner is first identified by the system. During the signup process, the system sets up all the static information, the RWV of the learner based on his place of origin, adds that new information to the DB and communicates it to the appropriate running agent so that it be taken in account for any further decision.

Steps:

- Signup: The Browser sends [username, password, placeOfOrigin] to the proxy.
- Create-user: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy agent forwards [username, password, placeOfOrigin] to the Culture Modeler.
- InsertUser: The Culture Modeler updates DB with information corresponding to the new user. This is also the step where the initial Cultural Student Model (CSM) is created. The CSM is sent back to Proxy.
- Signup-2-sleepy: The Proxy proceeds to inform the Allocator of the presence of a new user and forwards it her CSM.
- Allocate-user: The Allocator sends the CSM to a Sleepy with yet a free slot. If no sleepy is found free, a new one is created first. The Allocator also keeps track of which Sleepy keeps the CSM of which user.

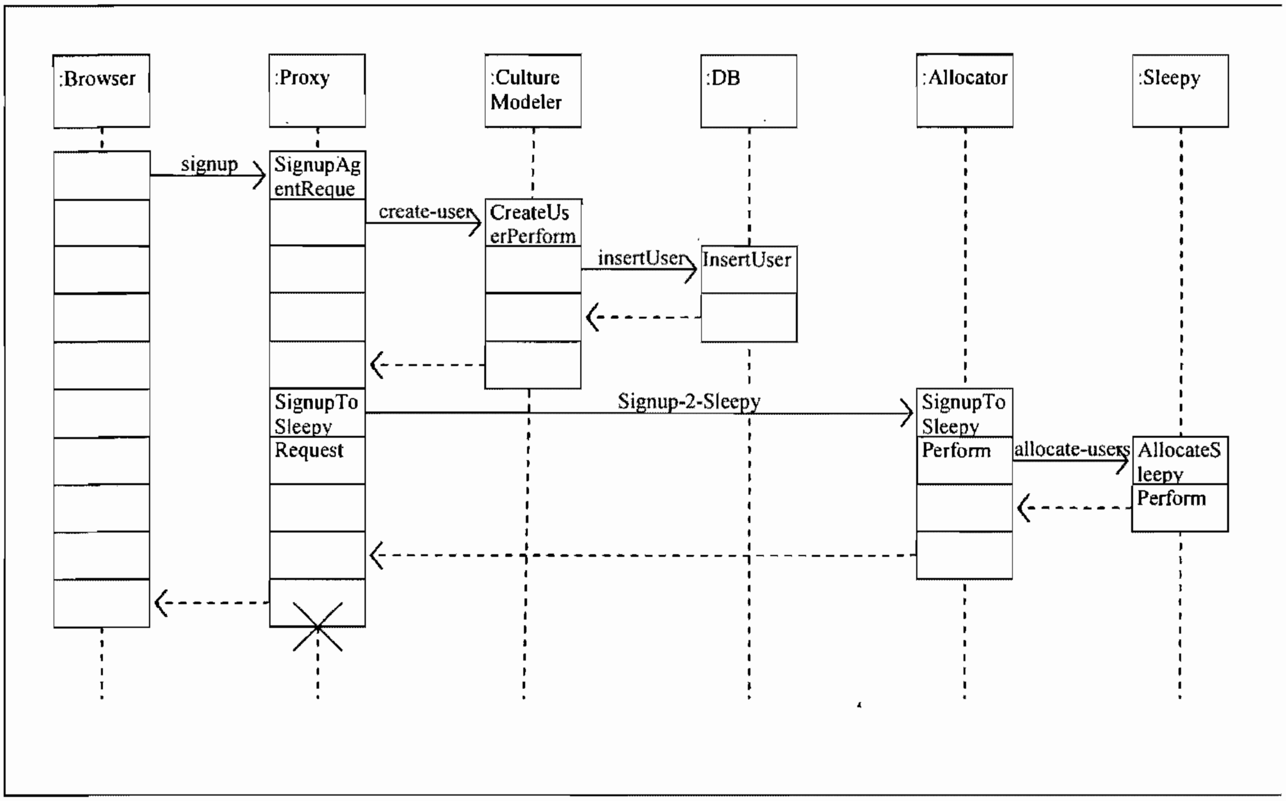


Figure 3.3 Signup sequence

3.3.2. Login

Process description:

Upon signing up, once the static information about the learner is collected, the learner can now interact with the system and start feeding it dynamic information that will be used for building her NRV. The login process mainly consists on creating a dynamic Client agent that will encapsulate the information about the learner.

The Client\s first action is to send the address of that learner's first page to display to the proxy.

Steps:

- Login: The Browser sends [username, password] to the proxy.
- Login-user: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy agent forwards [username, password] to the Allocator.
- UserExist: The Allocator checks in DB the validity of [username, password]. That is the only time when the Allocator will ever communicate with the DB. This operation is allowed because it does not affect the info in the DB.
- Create: If the user exists, the Allocator creates a new Client agent.
- Put-sleepy: The Allocator sends to the Client the ID of the Sleepy [SleepyAID] that is keeping the learner's Cultural Student Model (CSM).
- Load-model: The Client request the learner's CSM from the appropriate Sleepy, thus freeing one of that Sleepy's slots.
- Login-page: Once the proxy receives notice that the Client is up and running, it request it the address of the learner's first page to display.

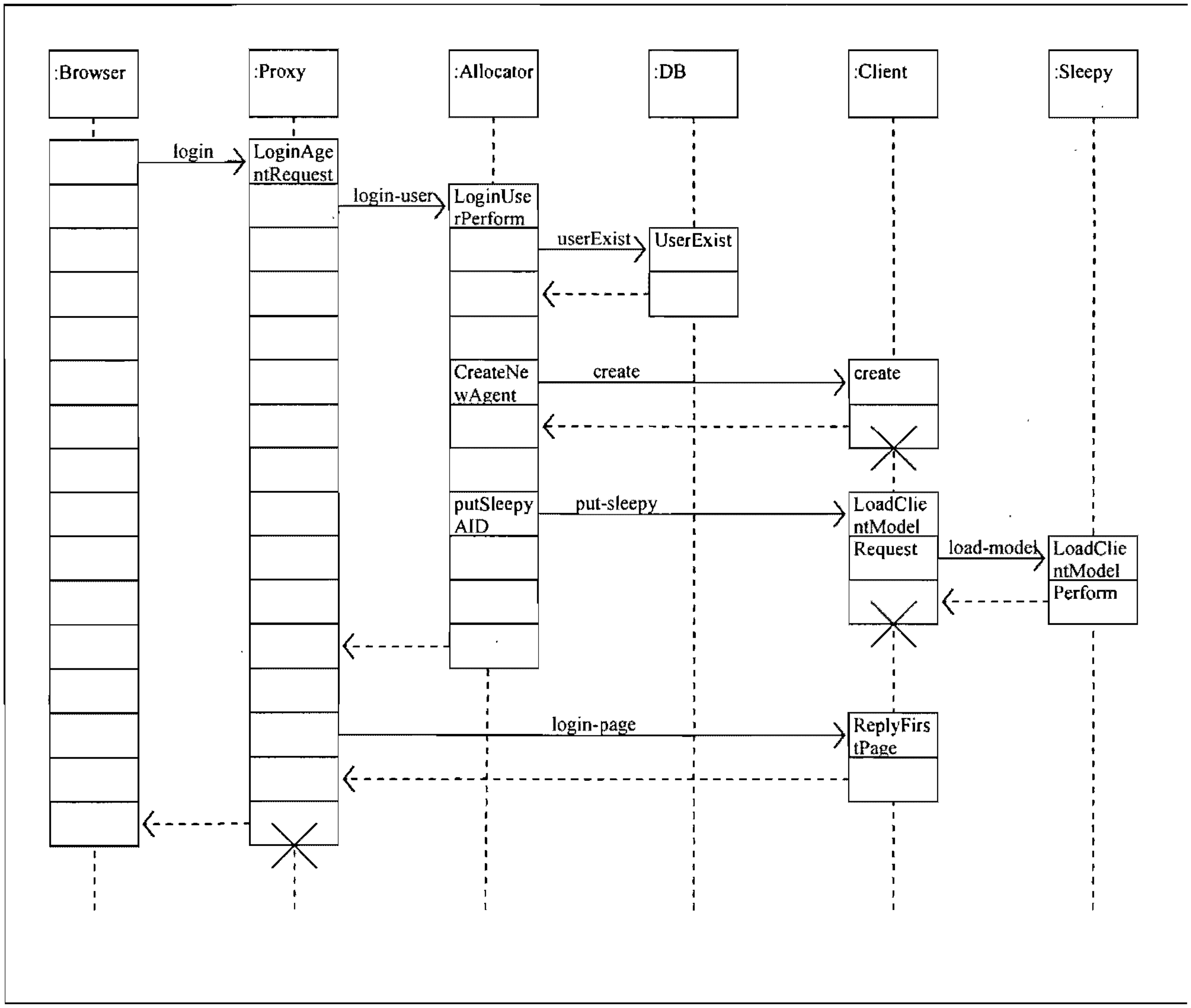


Figure 3.4 Login sequence

3.3.3. Rank Resource

Process description:

The learner can interact with the system in one simple way. That is by rating the resources provided on the page he is viewing. Those ratings are used to build the NRV that will be taken into account for making decisions on the choice of resource to display.

Steps:

- Rank: The Browser sends [username, resourceName, rank] to the proxy.
- Rank-resource: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy agent forwards [username, resourceName, rank] to the Culture Modeler.
- Rank-resource: The Culture Modeler send [username, resourceName, rank] to the DB. The DB updates the rank given by that learner to the Resource and sends back the updated CSM to the Culture Modeler.
- Change-model: The Culture Modeler sends [CSM] to the client to update its version of it.

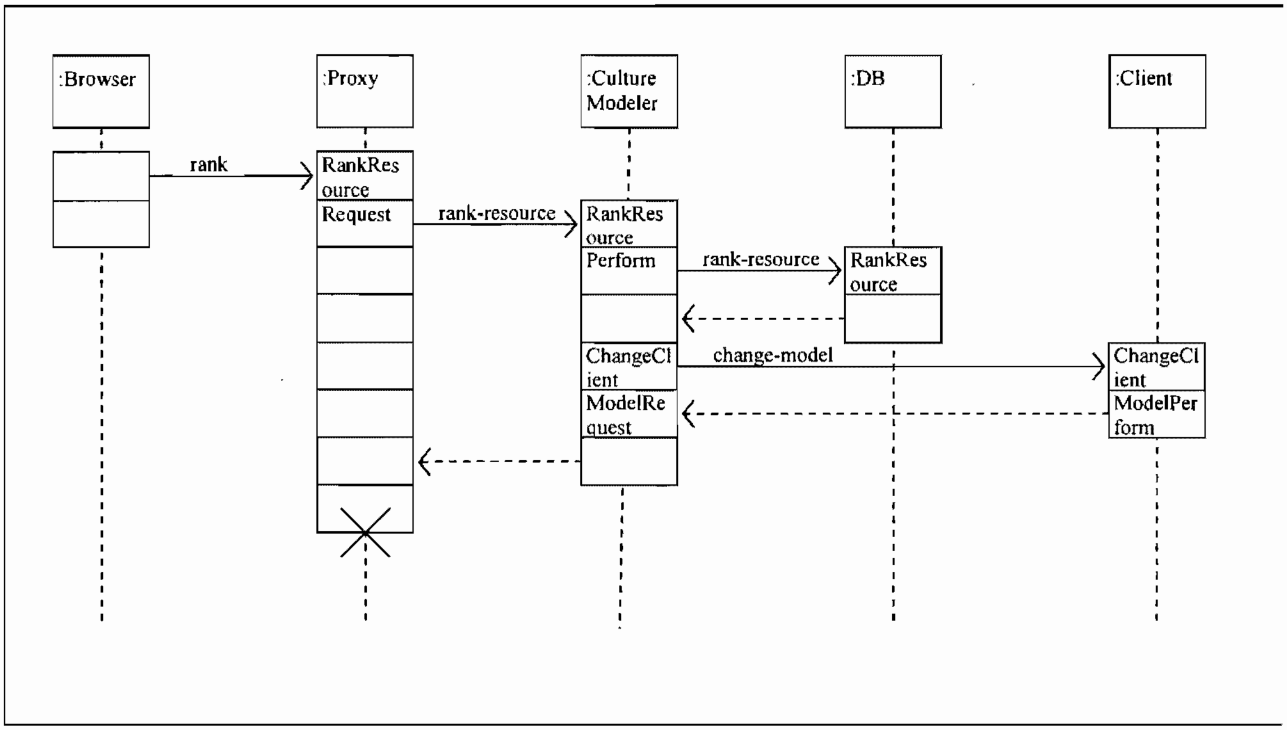


Figure 3.5 Rank resource sequence

3.3.4. Solve Page

Process description:

Before the system displays a web page, it uses the other users rating to choose the appropriate Resources to show on that particular page. This is the process through which that choice is made. The system first requests a list of the Notion present on the page to display. It then sends those Notions to all Clients signup as well as to the Sleepy that carries information for those users that are not signed up. They each suggest a Resource as well as sending back the value of their similarity to the requesting learner. The system picks the Resource suggested by:

- the most similar user based on NRV
- or based on RWV in case NRV is not computationally possible or yields a similarity number too low (less than 0.5)
- or randomly in case RWV is not computationally possible or yields a similarity number too low (less than 0.5)

Steps:

- Solve-page: The Browser sends [username, pageName] to the proxy.
- Build-page: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy agent forwards [username, pageName] to the Client.
- List-notion: The Client sends [pageName] to the Culture Modeler. The Culture Modeler forwards it to the DB and retrieves the name of every Notion present on that page. They are sent back to Client.

- Find-solver: For each Notion, the Client sends the [CSM, Notion] to the Allocators to find the best Resource for that Notion.
- Solve-notion: The Allocator sends each [CSM, Notion] to both the Clients for all logged in users and the Sleepys for all users that are not logged in. The Allocator receives the Resources that each Client has rated as well as those that the user that has the most similar CSM in each Sleepy has rated as best. Those resources are received together with the value of the similarity between the user and the Client that rated them. In other word the Allocator receives back a series of [bestResource, similarityValue]. This step ensures that the Clients do not have to disclose their CSM to the Allocator.
- The Allocator sends back the [bestResource] of the user with the closest [similarityValue] to the Client of the learner.
- Convert-File: For each Notion, the [bestResource] is given to the proxy which uses it to convert the Cultural Template into a script that that be understood by the Brower.

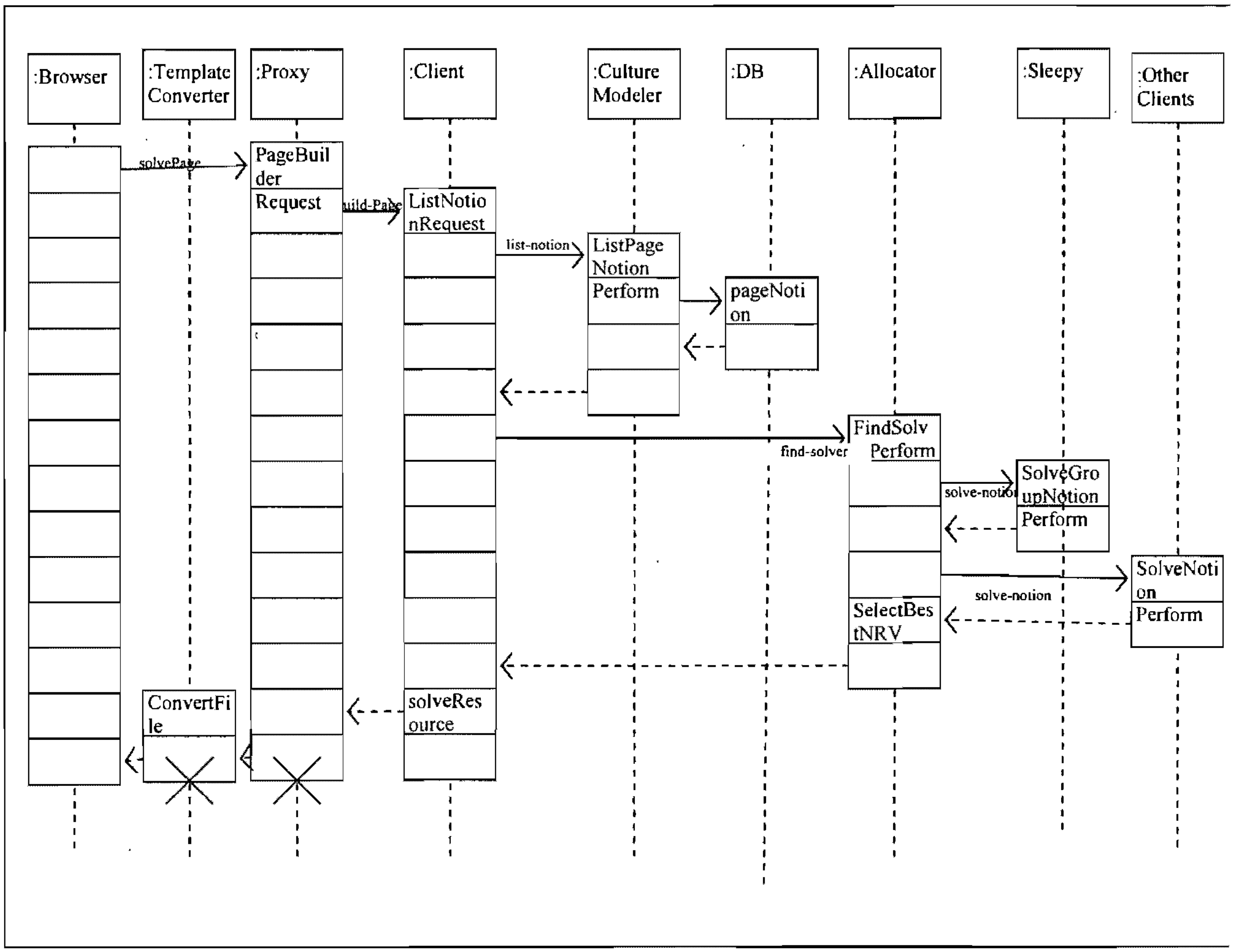


Figure 3.6 Solve page sequence

3.3.5. Evaluate Lesson

Process description:

This step is an attempt to validate (or invalidate) the beliefs of the leaner about which Resources best suits him. The system tries to add information on the learner's understanding of the lesson he has been taught. The underlying assumption is that the better the understanding of the lesson has been, the more appropriate were the Resources that were chosen to illustrate that lesson. The "understanding" can be determined in any manner (such as quiz is given at the end of each lesson). The rank given by the learner for each resource during the course of a lesson is weighed by a coefficient derived from the level of that "understanding" (such as the grade scored on the quiz).

Steps Description

- Evaluate1: The Browser sends [quizName, answers] to the proxy.
- Resource-seen: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy request from the Client a list of all of the Resources the learner has encountered during the lesson [resourcesSeen].
- Evaluate2: The proxy uses the proxy agent to bridge the web server to the Proxy agent on the Java server. The Proxy agent forwards [quizName, answers, resourcesSeen] to the Culture Modeler.
- Evaluate-page: The Culture Modeler sends [quizName, answers] to the DB that compares it to the real answers and sends back the learners [score]. The [score] is used to compute the [understandingCoef].
- GetResourceNote: The Culture Modeler sends [resourcesSeen] to the DB and gets the current rank of thoses Resources. It uses the understandingCoef to update the rank of each of the Resources.

- Rank-resource: The Culture Modeler updates the rank of those Resources in the DB. The DB replies with the updated CSM of the learner.
- Change-model: The Culture Modeler forwards the updated CSM to the Client.

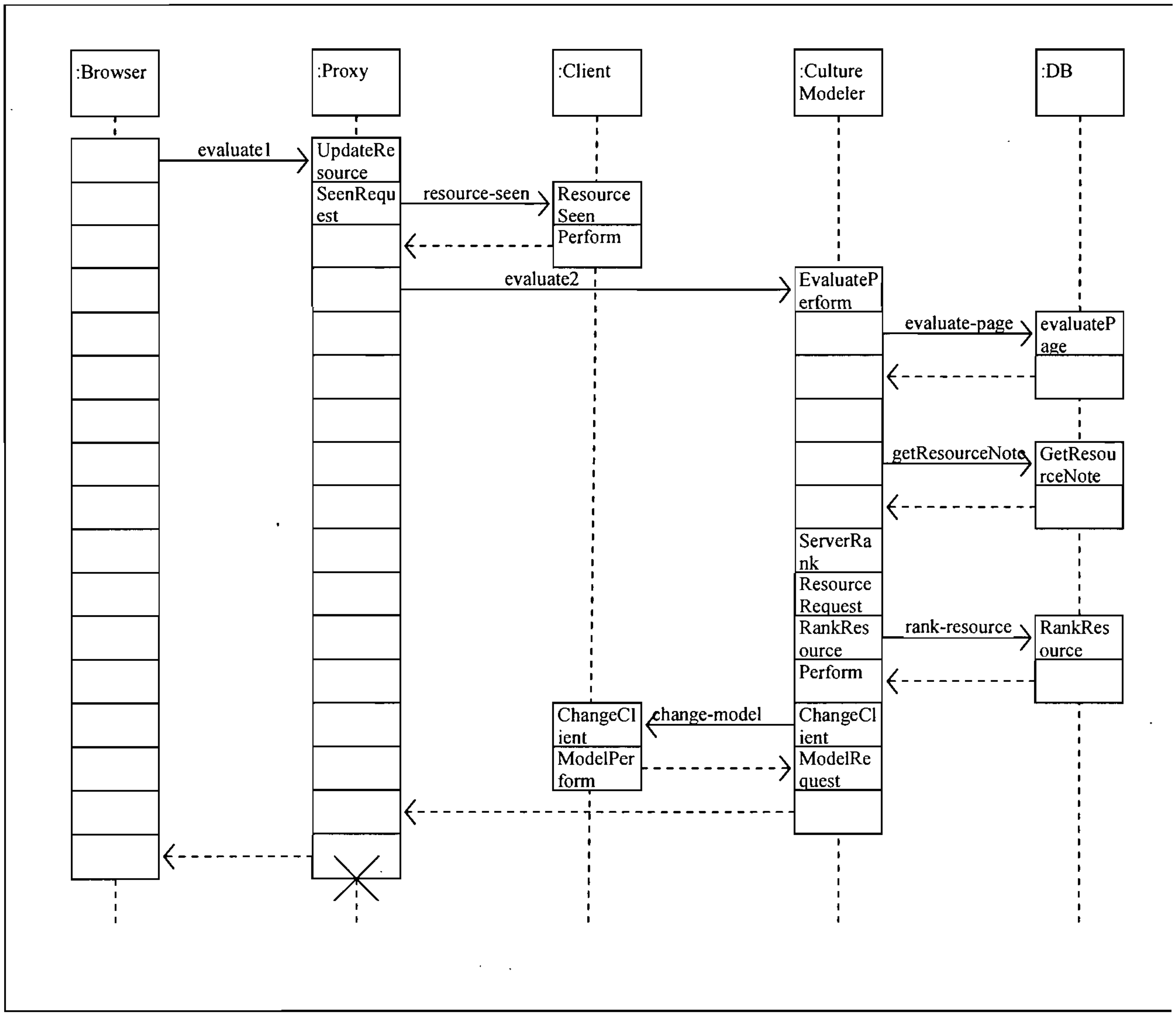


Figure 3.7 Evaluate lesson sequence

3.3.6. Towards implementing the main functionalities

Those five functionalities are those that must principally be implemented in order to have a functional version of CAMELEO. An implementation of CAMELEO would use them in the following manner:

- First a user would need to "Signup" in order for his RWV to be build.
- Once that step is completed the user could "Login" and that would active a Client agent that can use the info in the RWV.
- Whenever the user would request a new page of the lesson, the "Solve Page" functionality would be used to decide which resource should appear on the page.
- The user would have the choice to act upon the resources using "Rank Resource" and the ranks would be used to build up his NRV.
- Finally, at the end of a lesson the user would take a test which result would be sent with "Evaluate Lesson". The results will be used to further refine the CSM.

Following those steps, one can start building an implementation of CAMELEO.

Chapter 4: Implementation

4.1. Description

In order test the CAMELEO architecture described in Chapter 3 we decided to build a system that would gather the type of information that could be used to put together a Cultural Student Model (CSM). That CSM could later be used by an ITS to make decision far more complex than the ones our system currently does.

Our implementation of CAMELEO works together with a web based tutoring tool that emulates the most basic functionalities of an ITS. Its main purpose is to teach a lesson to a given student after which it quizzes the student on the knowledge acquired in the lesson. It must be clear that the tool only emulates the superficial functionalities of an ITS as it is not a true ITS. It is not build according to any specification of ITS architecture, for the focus is not on the tutoring tool. Yet it acts as an ITS would and could be easily substituted for an ITS.

CAMELEO that we have defined as a cultural filter works on the external form of the web pages presented to the student by the tutoring tool. This means that it acts upon the choice of the resources used to illustrate each page. In this implementation, resources are understood to be only images and sounds. CAMELEO selects the images and sounds that should appear on each page. That choice is based on the users CSM and its motivation is to build the page that will best help the learner understand the notions that are presented on the page. In other words it is a web based tutoring tool that adapts its images and sound clips to the students CSM.

The whole system must however be culturally intelligent. How can we guarantee that it is? Cultural intelligence is described by Earley and Mosakowski [2004] as a "seemingly natural ability to interpret someone's unfamiliar and ambiguous gestures the way that person's compatriots would". Two different aspects of the concept of cultural intelligence are needed in a CAMELEO: Interpretation and Adaptation.

- Interpretation corresponds to the system building the students CMS. This is done in a first step where the student is asked to answer a series of multiple choice questions on images and sounds.
- Adaptation is the step where the system actually uses the information gathered earlier. This is done during the presentation of the lesson. The student follows a lesson that is adapted to the CSM build earlier. The interpretation phase is however never interrupted, even after the adaptation phase has started. This guarantees that the system continues to learn from the reaction of the student to the choices it is making.
- We added an extra step that is a quiz. Its purpose is dual. The first is to complete the emulation of a tutoring tool. We ask question to assess the understanding of the lesson and give a grade to the student. The other purpose is to use that grade as a measure of the student's understanding of the lesson but also in order to validate the choices of resources by using a value that is more pedagogical than only the student's opinion.

4.2. Interpretation of the user's responses: building the CSM

The problem, in the case of this implementation, is to choose a resource (sound or image) to illustrate a text. The interpretation part happens when the system gathers information about the learner in order to build up the CSM. Theoretically, this step happens simultaneously as the adaptation.

In order to better explain this, let's take the concrete example of a real classroom environment. Here the tutor and the student are both humans. The tutor is trying to teach student a particular lesson containing a set of notions. The tutor has many ways of illustrating a notion, but which one to choose? The answer is dependant on the student. According to responsive teaching, the strategy will depend on whichever information the tutor possesses about the student cultural background.

The only information that the tutor possesses is the student's place of origin (which is provided at sign up). Therefore the tutor should try an illustration that is meaningful to a student of that place of origin. On the basis of the student's response to the tutor's choice, this one will learn more about the student and the next time he will use that new information. That is why interpretation and adaptation should theoretically occur simultaneously.

Yet in our case, and for the purpose of this test, we noticed that it is a lengthy process to build a CSM in that way. It eventually yields results, but we are limited by the time a student is willing to spend testing our system. We had to find a way to quickly build a CSM, even if a small one, before starting the lesson.

The solution chosen is to precede the lesson with a phase where the system gets acquainted with the user's preferences. Interpretation means learning to know the students reaction to some choices of illustrations (or resource) for particular notions. For each notion, the student must rank the level to which the illustration chosen relates to the notion (Figure 4.1).

The system builds a profile based on association between resources and the values and emotions associated by the user to those resources.

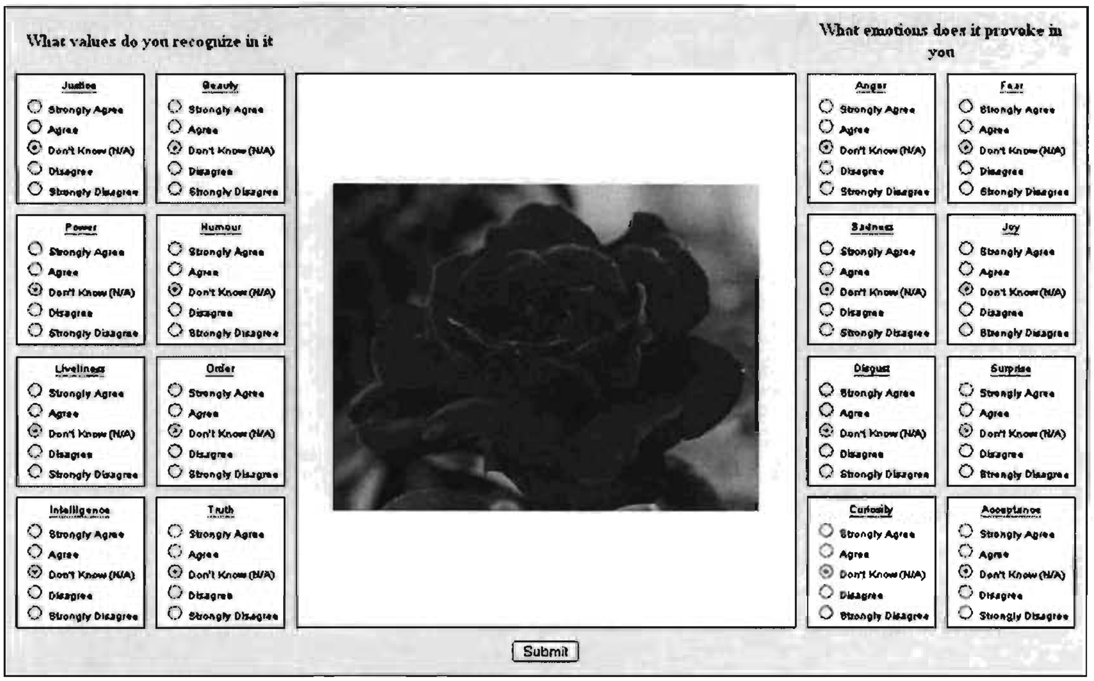


Figure 4.1 Survey page

The challenge here is to build a CSM that will quickly succeed at distinguishing between students in the way that they react to images. Cultural researcher Raymond Williams wrote in 1958 that culture is a "*set of distinctive spiritual, material, intellectual and emotional features of society or a social group, and that it encompasses, in addition to art and literature, lifestyles, ways of living together, value systems, traditions and beliefs*" [Williams, 1958]. A 2002 article by the United Nations agency UNESCO quotes this definition and agrees with it.

We will concentrate on the terms "emotional features" and "value systems". The interpretation step focuses on determining the extent to which a particular image or sound relates to the students "values systems" and "emotions". The student is asked to rate the pertinence of a set of image and sounds (which are the resources) at representing some values and provoking some emotions (which are the notions). The following step is to give a useful definition of values and emotions.

The Values

In order to select the appropriate set of discriminating values, we chose to turn toward Harvard professor Abraham Maslow suggestion that "*human beings are all born with an innate sense of positive and negative being-values. We are attracted to positive being-values such as: justice, honesty, truth, beauty, humor, liveliness, power, order and intelligence*" [Maslow, 1962]. That is the list from which we will choose the values that need be illustrated. We picked eight of those values. Our choices are: Justice, beauty, power, humor, liveliness, order, intelligence, truth.

Honesty was left out because we feared that it would cause confusion if listed along with truth. Those two values seemed to relate to concepts to similar in the eye of the general public.

The emotions

An identical approach was used in selecting a group of emotions that would encompass the range that best describe an individual. We will not go into too much depth about the definition of emotion because it is not the topic of this master. Emotions were simply used as a discriminating factor in a cultural frame. Additionally, the field of research dealing with emotions and its various definitions is vast and expands way beyond the scope of our work. Therefore, our choice of a list of emotions was made according to the subsequent research, not because of its pertinence but rather because it was a landmark in the research. Indeed, one of the most influential classification approaches in the study of emotion is Robert Plutchik's eight primary emotions. The emotions that Plutchik lists as primary are: Anger, fear, sadness, joy, disgust, surprise, curiosity and acceptance [Plutchik, 1980]. In this case, all eight of them were selected because there did not seem to be any confusion occurring in differentiating between them.

The ranking

Once the set of notions is determined and the images are chosen, we face the problem of evaluating the student's response. We need to associate a value to the association between Notions (emotions and values) and Resources (images and sounds). One of the questions that the student is asked in Figure 4.1 for instance is: "What values do you recognize in the image of the flower (rose)?" One of the suggested answers is: "Beauty". We used a five-point Likert scale [Likert, 1932] to gauge the student's appreciation. A five-point Likert scale is a type of psychometric response scale often used in questionnaires. It is the most widely used scale in survey research. When responding to a Likert questionnaire item, respondents specify their level of agreement to a statement. The system converts Likert's qualitative values into scalars that can be used for computation as illustrated in Table 4.1.

Table 4.1 Conversion of Likert scale into quantitative value

Likert qualitative answer	Quantitative value
Strongly disagree	1
Disagree	2
Neither agree nor disagree	3
Agree	4
Strongly agree	5

The student goes through the list of all images and sound clips and indicates the level to which the Resource relates to each one of the Notions on the page. That is a quick and efficient way of building a CSM that is not solely based on the place of origin, meaning it is not only based on the RWV but also on the NRV of the student.

Thus prepared we can move on to the lesson and the adaptation phase armed with more trustworthy weaponry.

4.3. Adaptation to the user's culture: using the CSM

The Adaptation phase is the step where the CAMELEO system will really be put to test. It is the phase of recommendation. As explained earlier this is actually the only necessary step of all. The previous one was merely a way of accelerating data acquisition. It is understood that Adaptation comprises Interpretation as both are occurring simultaneously. Each time the system makes a recommendation its outcome is analyzed and once again interpreted. Before all, let's first explain the Adaptation process.

CAMELEO is designed to work together with an ITS. In this implementation we did not build a complete ITS, in order not to lose focus, and concentrate our strength on the cultural filter rather. So we simply build a set of web pages that are meant to teach a lesson. The lesson contains multimedia elements such as sound clips and images. Those sound clips and images are meant to illustrate Notions. The source of those images and clips are our Resources.

In other words, the layout of the pages is as follows. Each page contains a short text, generally one or two paragraphs. Additionally, each page contains a space for inserting an image or a sound clip. That image or sound clip is meant to illustrate a Notion discussed in the text.. For each Notion there is a set of possible images that could all be used to fill the space. The issue is to choose the one that best fits the student at hand. Finally, we give the student the possibility to rate the choice of illustration, using the five-point Likert scale, as the lesson is going on. The page layout is as shown in Figure 4.2 for two different students. As one can see, the layout does not vary but the illustrations do. They depend on the culture of the student.

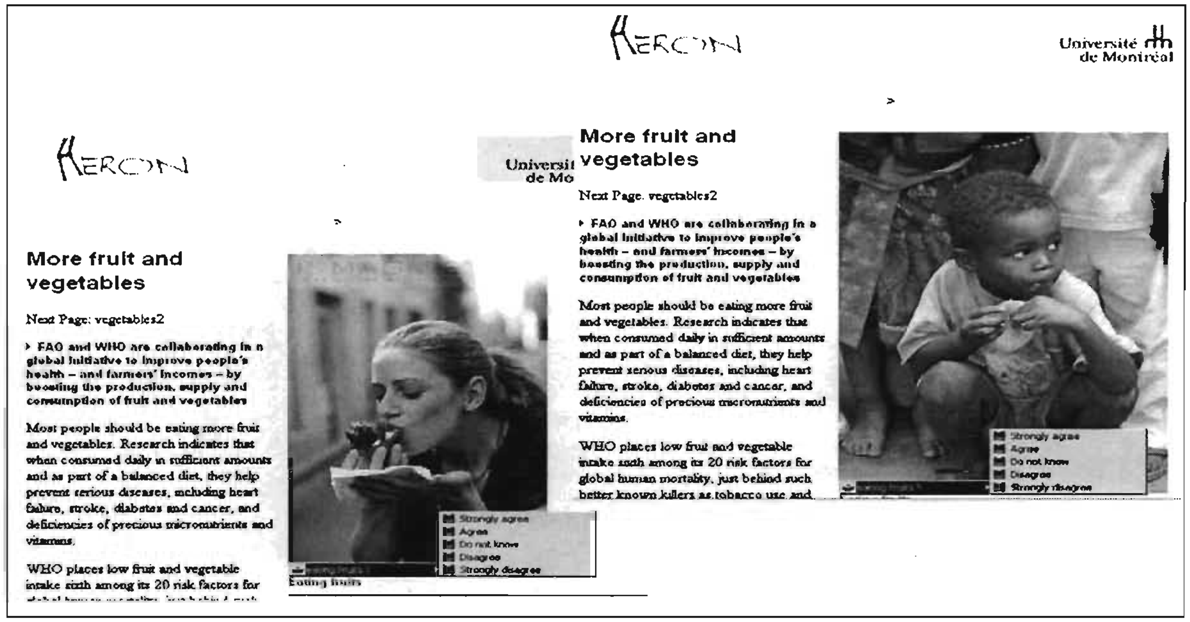


Figure 4.2 Lesson page

The text

For this implementation of CAMELEO we wanted to select a lesson that would touch universal topics and at the same time could be understood differently depending on the cultural background of the student. The universality is a necessity to ensure that all of our subjects understand, or rather *have their understanding* of the topic. The topic must also be interpreted in different ways so it must be about an issue that is faced independently by different populations and not specific to one in particular.

Therefore, we chose for the body of our lessons an article from [FAO, 2006]. The topic is on the consumption of fruits and vegetables over the globe. It discusses the dangers that can arise from the lack of consumption of fruits and vegetables, as well as the reason why there is such a problem. Each page is a short paragraph excerpt from the main article.

We take the student through snippets of the article one page at a time. The original article is split up over a total of six pages.

The illustrations

Once the article was divided into pages, we went through the task of selecting the main notions discussed in each page. Out of all those notions we picked one for each page and searched internet image databases (Google image, iStockphoto) for images that related to the Notions we picked. For each Notion we selected three Resources that we found representative and would speak differently to individuals from different cultural backgrounds.

This methodology incorporates the inevitable bias of a selection made by individuals sharing the same culture. Even though the choice of resource was approved by colleagues of different origins, it still remains that we all belong to the same laboratory and are working in the same country. Most of us are also engineers and computer scientist, so in a certain way we cannot sincerely claim the choice of illustration was made without a bias. Yet one could argue that it is the same methodology that would be followed in a real life situation. Experts gathering to set up an ITS would not necessarily be of a variety of cultures. They would simply use their judgment in deciding the set of illustrations that would speak to the widest range of students.

Cultural Template (CTP)

A cultural template (CTP), as described in section 3.2.3, is a general template of a page and is to be adapted to any student, on the basis of his cultural background. It must contain all of the pedagogical information. It must also leave spaces for variables when it comes to describing the Notions present on a page.

The implemented CTP (Figure 4.3) is adapted from HTML. Whenever an image or sound is culturally dependant, its source is described using *CTP tags*. Those tags will be converted downstream by the Template Converter into the right type of script. In our implementation, the web server manages JSP pages, therefore the CTP tags should be converted into JSP script. The format of the tag is as follows:

<CTPTAG>Name_of_Notion<CTPTAG>

In our web base application three CTP tags have been implemented. They are as described in Table 4.2.

Table 4.2 Description of CTP tags

CTP Tag	Type	Description
<CTP: IMG>	Image	The source of the image is culturally dependant
<CTP: SND>	Sound clip	The source of the sound clip is culturally dependant

Figure 4.3 is an instance of a CTP. In this example the Notion is "market". The Template converter will use this CTP together with the result of the Adaptation process and generate a JSP page with the right source for the image.

```
<h2>Contents</h2>
</div>
<ul>
<li class="toclevel-1"><a href="#Function"><span class="tocnumber">1</span> <span class="toctext">Function</span></a></li>
<li class="toclevel-1"><a href="#Types_of_Markets"><span class="tocnumber">2</span> <span class="toctext">Types of Markets</span></a></li>
</ul>
</td>
</tr>
</table>
                    </td>
                    <td height="96" width="292">
                    <p align="right">
                    <CTP:IMG>market</CTP:IMG></td>
                </tr>
            </table>
<p><script type="text/javascript">
```

Figure 4.3 Cultural Template

Live ranking

Once the Resources are selected and displayed to the page, the student can enjoy the lesson. Yet he might not agree with the decision made by the system. Let's consider the example given in Figure 4.2 . The image is meant to represent two possible illustrations of the Notion "eating fruit" for two different students. The system assumes that one or the image is the best Resource for representing that Notion in the case of one or the other student. This does not mean that it is. One needs a way to verify the suggestions of the system. CAMELEO is based on a constructivist approach to learning. It means that we rely on the student's interpretation of the Resources we are presenting him with. We must allow the student to respond to the suggestion. The student must be able to confirm the fact that the picture is to him truly representative of the Notion.

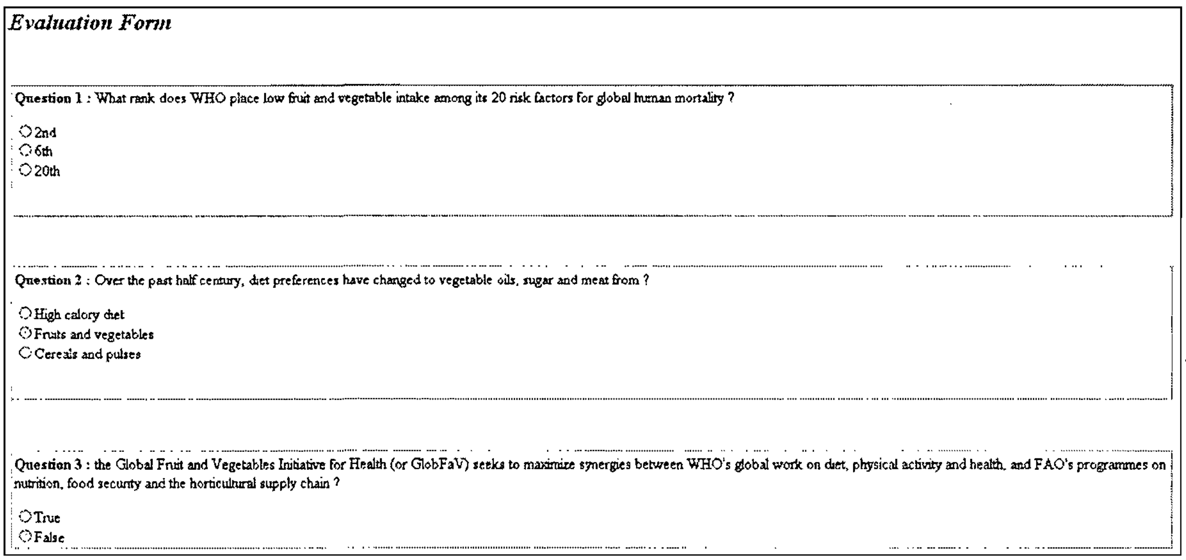
Before that is possible, the student has to know which Notion the resource is meant to represent. Secondly, he must be able to rate his level of agreement. To make those things possible we built a JavaScript *Live Ranking Menu* (Figure 4.2) that appears under the image or the media player (in the case of a sound clip), when the mouse cursor moves over those Resources. The menu indicates the name of the notion that the resources represents, and also contains the five-point Likert scale. The student can thus rank the level to which he agrees to the suggestion.

Upon ranking, the system reevaluates the CSM and instantly recommends a different resource based upon the new profile. In this manner, we provide the student with a dynamic way of interacting with the system.

4.4. Quiz

The methodology as it is described so far builds the CSM solely on the basis of the student's opinion of the systems suggestions. Indeed it is a close enough emulation of a constructivist approach to teaching. Only, a deeper question arises. Is the simple fact that the student "believes" that a particular Resource is the best for his understanding of a lesson enough to consider when building a lesson? What is a pedagogical value of such an application? Or furthermore, how does one verify that pedagogical value? What if the student is wrong and the resource he believes best is not the one that would best help him understand the lesson?

What better way to check the student's understanding of the lesson than to test him about it? At the end of the lesson session, a quiz is conducted in order to evaluate the level to which the lesson was understood (Figure 4.4). The quiz is presented in the form of a multiple choice questions form (Appendix B.1). Each question applies to one of the six pages of the lesson. There is, therefore a total of six questions for the whole lesson.


Evaluation Form

Question 1 : What rank does WHO place low fruit and vegetable intake among its 20 risk factors for global human mortality ?

- ○ 2nd
- ○ 6th
- ○ 20th

Question 2 : Over the past half century, diet preferences have changed to vegetable oils, sugar and meat from ?

- ○ High calory diet
- ○ Fruits and vegetables
- ○ Cereals and pulses

Question 3 : the Global Fruit and Vegetables Initiative for Health (or GlobFaV) seeks to maximize synergies between WHO's global work on diet, physical activity and health, and FAO's programmes on nutrition, food security and the horticultural supply chain ?

- ○ True
- ○ False

Figure 4.4 Quiz page

The questions have been designed with two purposes in mind. The first is obviously to reflect the general theme discussed in each page of the lesson. One should not forget that we are trying to check whether or not the student understood the lesson as a whole. Yet, we are also trying to use the quiz as a way to determine whether the system is taking the best choices. Therefore we geared the questions towards the themes illustrated by the Notions that where adapted using CAMELEO. Thus we can suppose that the illustration helped in answering the questions. This is the big hypothesis that this steps builds upon. The hypothesis is that the better the choice of Resources the better the understanding of the Notion and therefore the better the score obtained in the quiz.

The score obtained through that quiz are to be used to weigh the ranks given to every resource encountered during the lesson. This is achieved using the following process.

- First the score obtained on the quiz is normalized over five in order to be comparable to the five-point Likert scale.
- Then the following operation is performed on the value of the rank given to each Resource seen:

$$N = \frac{O + Q}{2} \tag{3}$$

Where O is the old rank given of the resource, Q is the normalized over five score obtained on the quiz and N is the new rank of the Resource.

This can cause a big shift between the student's appreciation of the suggested Resource and the actual value that it will end up having, but we understand the score to the quiz to be of an importance fundamental enough to justify this weight.

Chapter 5: Experimentation

5.1. Methodology

The principal aim of our experimentation was to gather data on learners' appreciation of the resources that CAMELEO presents them with, to compare the system's choices with the learners' actual preferences in the course of an e-learning session. We wish to verify two hypotheses.

- The first is whether CAMELEO is able to provide a learning environment that is culturally meaningful for the user of our system.
- The second goes somewhat beyond the scope of our study, but is an interesting one to verify: we wish to know whether, if CAMELEO can teach according to the request of responsive teaching, then that does truly affect students' results.

The system implemented was put on a public server accessible over the internet. 117 users were asked to sign up providing their country of origin. Following that step, a RWV was generated for each one of them. Over the course of a learning session, a NRV was built and completed for each student.

Because of the setting being online, students where unsupervised. They followed the experiment from their homes or any place where they had internet access. Even though the specifications were clear, it cannot be guaranteed that they were followed. For instance we could not guarantee the information provided by the student about their place of origin. We could not guarantee that the quizzes were not taken by groups of individuals. We also did not keep track of the time taken to complete the lesson. Although those points prevent us from having real control over our testing group, we feel that they would nevertheless give our experiment more credibility because we effectively reproduce the setting of an e-learning course.

The points that we did keep track of were the following:

- The experimentation was conducted over the course of two and a half weeks.
- The number of the participants was 117.
- The countries of origin of those participants were diverse. According to the Hofstede's classification used for building the RWV [[Hofstede, 2001] they were from Western Africa (40 participants), France (28), United States (10), Arab World (9), Canada (6), Eastern Africa (6), Spain (4), Taiwan (4), England (3), Germany (3), Colombia (2), Brazil (1) and Russia (1).

5.2. Results and Interpretation

The experimentation is evaluated in two different manners.

The first is a self evaluation of the system that verifies its propensity to adapt not only to a user during the time of a session, but also the accurateness of its decision making over the course of the whole evaluation: the system's adaptability.

The second is an attempt to measure the positive impact that the methodology has on learning.

5.2.1. System's adaptability

We have defined CAMELEO as a system that adapts to an ITS student, and shows him the representation of notions best suited for his culture. The final aim is to help the student better understand the notions he is being taught. The key word is "adapt". In order to verify that the student results are influenced in any manner by the work that CAMELEO is performing, we need to first insure that CAMELEO is performing any work at all, and if so, that it is performing the appropriate work. Does the system behave the way that was planed? The first test is one on the system. Does it really adapt to the student, and in which way does it do so?

Adaptation can be measured by observing the system behavior over time and checking whether there is any sign of intelligence in it. In order to do so, the following test was performed: Following the system modus operandi, every time a new page is displayed for the student, CAMELEO uses the students CSM to determine the resources that best fit him. Track was kept of those choices. That allows us to know the system's first choice at the moment the page is presented. Once the page was presented, the user was given the opportunity to rank the resources that he saw on the page. The system worked out another computation and gave another suggestion. No track of those subsequent suggestions was kept. That was not necessary. The student could go on and rank again the new resource, the system made new recommendations and so on. Eventually the student chose to move on the next page. The whole process started over. Our first test was to check whether the system's first choices were in concordance with the students' best ranked resource. Or rather it was whether, over time the system's first guess and the students' best choice tended to converge. Did the system's and the students' choices match, could the system adapt to the student? Figure 5.1 shows the percentage of matches over time as we move from one user to another. For each user the value of that percentage or M was computed as such:

$$M = \frac{100S}{S+F} \tag{4}$$

Where S is the number of successful matches between the system's first choice of a resource and the user best rated resource for the same notion and F is the number of failed matches. Those computations are based on the values found in Appendix B.2.

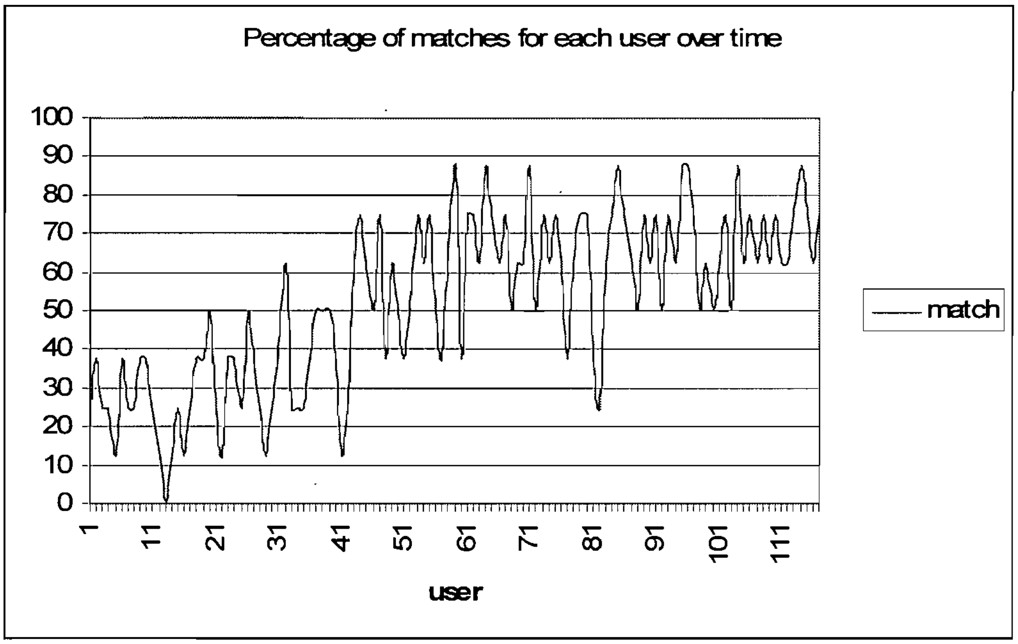


Figure 5.1 Percentage of matches for each user over time

The results shown in Figure 5.1 can be interpreted as follows:

Generally, over the course of the whole experiment one notices a significant increase of the percentage of matches. Starting from around 30 per cent, they eventually end around 70 per cent.

More specifically, the graph can be broken down to two periods displaying different types of behavior. The first period is from user 1 to the vicinity of user 45.

- That period starts at a rate of one third of successful matches: considering the fact that the system chooses between three resources, that number is a good indication that most decisions are first taken at random.
- The trend is to an increase: This is the period where the system comes in contact with the first users. We have a low peek at 0 towards the beginning and a high peek at 70 toward the end. The systems suggestions converge towards the students best ranked resources.

After that first period where the system learns to adapt to users of various backgrounds, there is another period, showing a different type of behavior.

- From around user 45 until the end of the experimentation, the general trend is stable with a mean that shift from around 60 to around 70 per cent. That result can be interpreted as the system having a better understanding of its users' expectations. It has taken only 45 users to reach stability. That number is probably not meaningful for it must depend on the pool of users tested.
- The second period's maximum peeks climb up to 90 per cent of successful matches. The minimum peeks are around 40 per cent except for one student that gave a result of 25 per cent. Therefore, except for that single data point the matches oscillate between 40 and 90 percent again with a mean that starts around 60 and quickly rises to 70 percent.

A first quick increase and then stability is the behavior expected from a system that learns and adapts. This observation shows that the system quickly adapts to the pool of users that participated to the test. The increase is a good sign of performance but the later stability is more ambivalent. It can be interpreted in two manners.

- It can be seen as the system reaching stability, in which case best performance would be around 70 percent.

- It can also be interpreted as an increase at a much slower rate that could only be seen on a much bigger scale. An experiment involving a greater number of participants would be necessary to test that point. But for the purpose of our study, the results obtained show a very encouraging trend.

As a general rule it can be said that CAMELEO can adapt to the users cultures, or that users recognize themselves in the choices that the system makes for them. If the "The academic achievement of ethnically diverse students will improve when they are taught through their own cultural and experiential filters" [Au & Kawakami, 1994; Gay, 2000], then CAMELEO efficiently provides that cultural and experiential filter.

5.2.2. Impact on learning

Once it has been verified that the system correctly adapts it predictions to the users' cultural background, we need to verify another point. The aim of CAMELEO is to improve learning. It is obvious according to our previous results that users recognize themselves in the choices that the system makes for them. The setting once in place, the idea was to try and verify the hypothesis that, in our case, "the academic achievement of ethnically diverse students will improve" [Au & Kawakami, 1994; Gay, 2000]. This is a more difficult task to carry out. In order to verify the hypothesis, we perform the following test: At the end of each student's session, a quiz is given. Then answers provided to the questions (Appendix B.1 for Lessons and Questions) are graded and the grades are converted to a scale of 100. Figure 5.2 shows the results of those quiz from the first to the last user.

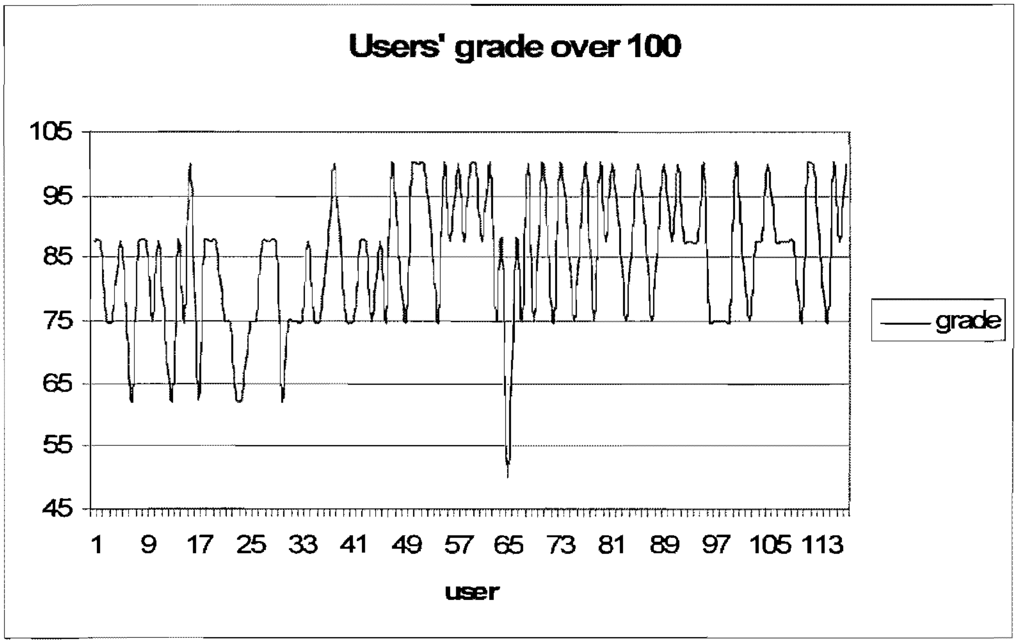


Figure 5.2 Users' grade over 100

The results can be interpreted as follows:

In general, the grades are good, ranging from 62.5 to 100. We notice a timid trend to an increase between the beginning and the end of the experiment. Once again the graph can be broken down in two periods of time.

- The first period runs from user 1 to the vicinity of user 40. The grades have a low peek of 65 per cent and a high peek of 87.5 per cent, except for one data point at 100 per cent.
- The second period, from around user 40 to the end show better grades, with the highest at 100 per cent and the lowest at 75 per cent. Once again there is a data point that does not follow the trend and drops to 50 per cent. That data point is so isolated that there is an urge to disregard it.

Although one could feel an inclination to conclude that the grades increase over the course of the experiment, it is better to be careful. There are points for and against that that hastened conclusion.

- One point for that conclusion is that indeed the means of the grades increase at least between the first and second periods determined.
- The second point is that those periods overlap with the periods found in section 5.2.1. This pushes us toward thinking that once the system finishes adapting, students grades improve.

Now there are a few things to consider before lapsing to conclusions.

- The first point is that the means increase from around 75 per cent in period one to around 85 per cent in period two. One could feel that 10 per cent is a poor increase at least too small to indicate that there is indeed a significant change in the results given by the experiment.
- The second point, apart from the fact that the grades are generally good and the questions asked generally easy is that it is hard to tell whether thêre should really be a correlation between the resources displayed and the users' understanding of the questions. The user probably understands the lesson better if better resources are chosen to illustrate it, but does he better understand the question? Nothing is more uncertain. In other words one cannot assert conclusively that the questions effectively test the accuracy of choice of the resources.

Although there is an increase in grade over the course of the experimentation, it remains hard to verify that CAMELEO has a true impact on learning. There need be further testing in order to ascertain that hypothesis.

Yet the fact that the hypothesis can hardly be verified with that experimentation does not mean that CAMELEO has no impact on learning. The "implicit rating" discussed in section 2.3.1 ensures that grades are taken into account when making the choice of resources. Resources yielding better grades are favored. This enforces that the student will be taught with the idea of academic achievement in mind.

If CAMELEO does not test the hypothesis, it nevertheless sets the grounds for testing it. In order to verify that the system not only creates an environment beneficial for learning, but also that the environment indeed improves academic achievement, one would need to consult specialists when building the quiz questions. Their task would be to grant that the quiz given at the end of a lesson really tests the resources. Although possible, such experimentation goes beyond the scope of this study. But once again, the grounds are set.

Conclusion and discussion

A system was presented that would allow us to create an e-learning environment adapted to students. That adaptation is based on the cultural background of the students. Indeed, pedagogical methods around the world vary enormously. Student from different places around the world learn in different manners. That variation is expected and can be related to their immediate environment. Yet, no matter where they are located, those students can have access to new technologies such as the internet. The internet is everywhere and students from around the world should be able to use and benefit from it. Yet, most of the knowledge source present on the internet has been developed by experts of western civilization keeping in mind the benefits of students from those same areas. The effort to open up the material to other cultures is a very noble one, but if the material is presented as it is, that would lead to a huge deal of misinterpretation. Misunderstandings lead to future mistakes. That is why we found it necessary to adapt e-learning environment to each single one of those students.

There are unformulated principles that guide the way in which each culture teaches their students. We know that those principles exist because teachers around the world are trained differently according to the population they are trying to teach. Only, those guidelines vary enormously and many of them are subjective or implicit. Besides, some of those rules require a very close acquaintance with the student. Some teachers base their relationship with students on his family background. Some base it on other factors. But all rely on the students' reactions and responses. They also frequently question the student's interpretation of the concepts they present him with. That general methodology is the one we chose to follow in order to adapt e-learning environment. As far as culture is concerned, the general consensus is that "the academic achievement of ethnically diverse students will improve when they are taught through their own cultural and experiential filters" [Au & Kawakami, 1994; Gay, 2000]

Our Cultural Adaptation Methodology for E-Learning Environment Adaptation (C.A.M.E.L.E.O.) is based on that philosophy, also known as Culturally Responsive Teaching.

The very challenge of this work was to find manners to "guess" or establish the cultural background of a student.

We turned toward student modeling and realized that student models were missing any form of cultural characterization. Before we can adapt teaching according to students' cultures, we need to add elements of culture into the student model. This led us to defining a component of a student model. The *Cultural Student Model* (CSM). A cultural student model was defined according to several definitions of culture. First, the most obvious definition was that of geographical localization. Students from the same country would initially be thought to respond the same way. We used Geert Hofstede's work [Hofstede, 2001] to initialize students' backgrounds according to their country of origin. That definition of culture is very limited and also outdated. Since media are growing more and more global, differences between individual are not so much related to their country of origin as it is to their vision of the world. The type of media they have access to and their interpretation of the information they see. That could be related to their social status, profession or any other factor. Culture is more complex. It is also "a process of production and reproduction of meanings in particular actor's' concrete practices (or actions or activities) in particular contexts in time and space" [Kashima, 2000]. We extended our methodology to that more profound definition of culture. The system creates CSM according to students' interpretation of the material they are presented with. Those CSM are used in a collaborative filtering process to create groups of cultures. A student will be taught with the same tools that worked on his most culturally close peer.

We implemented a system based on that methodology. Students online were asked to follow a lesson and grade the multimedia resources that are used in the lesson according to the accurateness to which those resources illustrate the notions they are meant to represent. In that way we have access to their interpretation of the teaching material. Intelligent agents manage those CSM and share the information needed to decide of the closest individual to any given student. The system tries out the resource that worked for that closest peer. The student's score of appreciation of that resource will again be used for future adaptation. The system is therefore constantly adapting CSM according to user responses and results.

Overall, the results we obtained were promising. One positive finding is that over time, the system we built adapts to the students background. After an experimentation involving 117 students, we were able to guess with as success rate of 70 percent the resource a student would estimates closer to the notions we present. We have also noticed that our methodology, based on collaborative filtering quickly reaches that rate of success.

We tried to put the "Responsive Teaching" methodology to test by checking whether CAMELEO actually yields better results on test score. Our findings here are more ambiguous. While tests scores increase over time, it is uncertain that their evolution is really significant and moreover that it is related to CAMELEO. Further testing must be done in order to clarify that point.

The main limitation of our system is the way in which we gather information about the student's interpretation of the material we show him. For instance, asking the degree (on a scale of 1 to 5) to which a particular resource represents a notion is a rather vague question. If the student is simply shown one resource how could he rate it? This has proved a very difficult task for our testing pool. A better methodology would be to ask the student to classify in order of increasing accuracy every resource that represents a notion. That would be a much better methodology but it would require that for each notion the student be shown every resource. That is an exercise that greatly diverts from the principal purpose of an e-learning session which is teaching material. We tried to answer that difficulty by creating a system that dynamically changes the resource if the student positively does not recognize it as properly illustrating the notion it is meant to.

The second limitation of the system is related to the first one. In order for the student to decide whether a resource properly illustrate a notion, we must explain to the student the notion we are trying to illustrate. This methodology is dangerous because we cannot guarantee that the student really understands the explanation. A written explanation is as much as source of misconception as any other kind. Also they must first understand the notion before deciding whether a resource illustrates it properly. They need to know the notion first. What if it is the notions that we are trying to teach? We should assume that the student does not know it before hand, and that the resource is a means to teach that notion. Would his understanding of the notion not already depend on the resource that illustrates it? The answers to those questions could probably be found in the studies about constructivism.

Future work would be to find an even more unobtrusive way of ranking resources according to students' interpretation. We need also find a way to produce more reliable CSM. Basically, future work should focus on tackling the limitations mentioned earlier.

Over the course of this project we have noticed one thing. It is that the problem of cultural diversity does not only relate to ITS. It relates to any type of media trying to present concepts around the world. Such media as online news websites could hugely benefit from C.A.M.E.L.E.O. The methodology could also be used to decide what kind of material would be potentially offensive to a given user. That would make surfing the web a much more benefic and pleasant activity.

References

Au, K.H., Kawakami, A.J. (1994). Cultural congruence in instruction. In E.R. Hollins, JE. King, & W.C. Hayman (Eds.), Teaching diverse populations: Formulating a knowledge base (pp. 5-23). Albany: State University of New York Press.

Aïmeur, E., Frasson, C. (1996). Analyzing a New Learning Strategy according to different knowledge levels. Computer and Education, 27, 115-127.

Appadurai, A. (1996). Modernity at Large. Cultural Dimensions of Globalization. Minneapolis: University of Minnesota Press.

Baeza-Yates, R., Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley, ISBN:0-201-39829-X, 1999.

Blanchard, E., Frasson, C. (2005). Making Intelligent Tutoring Systems culturally aware: The use of Hofstede's cultural dimensions. *In the Proceedings of the International Conference in Artificial Intelligence : ICAI2005*; Las Vegas, USA. June 2005.

Blanchard, E., Razaki, R., Frasson, C. (2005). Cross-Cultural Adaptation of eLearning Contents: a Methodology. International Conference on E-Learning, Vancouver, Canada.

Brown, J.S., Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, Vol. 2, pp.155-191.

Brown, J.S., Burton, R., de Kleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III, in: D. Sleeman, J.S. Brown (Eds.), Intelligent Tutoring Systems, Academic Press, New York.

Brusilovsky, P. (1994). The construction and application of student models in intelligent tutoring systems. Computer and System Sciences International, 32(1), 70-89.

Burns, H.L., Capps, C.G. (1988). Foundations of Intelligent Tutoring System : an Introduction. In Polson, M.V and Richardson, J.J (Eds) Foundation of Intelligent Tutoring Systems, p. 21-53, Lawrence Erlbaum Associates Publishers.

Burton, R.R., Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. Intelligent Tutoring Systems, Sleeman D. and Brown J. (Eds), Orlando, FL, Academic Press.

Chaffar, S., Frasson, C. (2004). Inducing Optimal Emotional state for learning in Intelligent Tutoring Systems. *In proceeding of International Conference of Intelligent Tutoring System (ITS'2004)*,Maceio, Brazil.

Chan, T.-W., Baskin, A.B. (1988). Studying with the Prince -The Computer as a Learning Companion. *Proceedings of the First International Conference on Intelligent Tutoring Systems, ITS'88* (pp. 194-200). Montreal, Canada.

Chan, T.-W., Chou, C.-Y. (1997). Exploring the Design of Computer Supports for Reciprocal Tutoring. International Journal of Artificial Intelligence in Education, 8, 1-29.

Clark, R.E. (1983). Reconsidering research on learning from media. Review of Educational Research, 53(4).

Conati, C. (2002). Probabilistic Assessment of User's Emotions in Educational Games. Journal of Applied Artificial Intelligence, Vol. 16, p 555-575

Devedzic V., Harrer, A. (2005). Software Patterns in ITS Architectures. International Journal of Artificial Intelligence in Education (IJAIED), Vol.15, No.2.

Dunn, P., Marinetti, A. (2004). Cultural adaptation: Necessity for eLearning. Learning in the New Economy e-Magazine (LiNE Zine)

FAO (2006). "More fruit and vegetables" Food and Agriculture Organization, http://www.fao.org/ag/magazine/0606sp2, May 27, 2007.

Foner, L.N. (1997) "Yenta: A Multi-Agent, Referral-Based Matchmaking System." *In First International Conference on Autonomous Agents (Agent'97)*

Frasson, C., Mengelle, T., Aimeur, E. (1997). Using Pedagogical Agents in a Multi-Strategic Intelligent Tutoring System. *In Proceedings of the Pedagogical Agents Workshop* (pp. 40-47), The 8th World Conference on Artificial Intelligence in Education, AIED'97, Kobe, Japan, August 18-22.

Frasson, C., Mengelle, T., Aimeur, E., Guarderes, G. (1996). An Actor-Based Architecture for Intelligent Tutoring Systems. *In Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS'96* (pp. 57-65). Montreal, Canada.

Friedman, J. (1990). "Being in the World: Globalization and Localization." In Global Culture: Nationalism, Globalization and Modernity. Edited by Mike Featherstone. London: Sage, 1990. 311-328.

Fuhr, N., Buckley, C. (1990). "Probabilistic document indexing from relevance feedback data". *In Proceeding of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 45-62, Bruxelles University.

Gay, G. (2000). Culturally responsive teaching: Theory, research and practice. New York: Teachers College Press.

Goldberg, D., Oki, B., Nichols, D., Terry, D.B. (1992). "Using Collaborative Filtering to Weave an Information Tapestry". Communications of the ACM, December 1992, Vol. 35, No12, pp. 61-70.

Goodman, B., Soller, A., Linton, F., Gaimari, R. (1998). Encouraging Student Reflection and Articulation Using a Learning Companion. International Journal of Artificial Intelligence in Education, 9, 237-255.

Hamburger, H., Tecuci, G. (1998). Toward a Unification of Human-Computer Learning and Tutoring. In B.R. Goettl, H.M. Halff, C.L. Redfield, & V.J. Shute (Eds.) Intelligent Tutoring Systems, Lecture Notes in Computer Science 1452 (pp. 444-453). New York: Springer-Verlag.

Heraz, A., Razaki, R., Frasson, C. (2007). Using machine learning to predict learner emotional state from brainwaves. *7th IEEE conference on Advanced Learning Technologies : ICALT 2007*, Niigata , Japan, July 2007.

Herlocker, J., Konstan, J., Borchers, A., Riedl, J., (1999). "An Algorithmic Framework for Performing Collaborative Filtering". *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230-237.

Hill, W., Stead, L., Rosenstein, M., Furnas, G.W. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. *In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, ACM Press, 194-201.

Hofstede, G. (2001). Culture's Consequences: Comparing Values, Behaviors, Institutions, and Organizations Across Nations, 2nd edition, London: Sage.

Johnson, W.L., Kole, S., Shaw, E., Pain, H. (2003). Socially Intelligent Learner-Agent Interaction Tactics. In U. Hoppe, F. Verdejo, & J. Kay (Eds.) Artificial Intelligence in Education (pp. 431-433). Amsterdam: IOS Press/Tokyo: OHM Ohmsha.

Kashima, Y. (2000). Conceptions of culture and person for psychology. Journal of Cross-cultural Psychology, 31(1), pp. 14-32.

Kautz, H.A., Selman, B., Shah, M. (1997). "Referral Web: Combining Social Networks and Collaborative Filtering". CACM, 40(3). 63-63.

Kelly, D., and Tangney, B. (2002): Incorporating Learning Characteristics into an Intelligent Tutor. *In: Proceedings of the Sixth International on ITSs, ITS2002.*

Kim-Prieto C., Fujita F., Diener E. (2004). Culture and structure of emotional experience. Unpublished Manuscript, University of Illinois, Urbana-Champaign.

Kolodner, J. L. (1993). Case-Based Reasoning. San Mateo, CA: Morgan Kaufmann.

Lang, K., (1995). "NewsWeeder: Learning to filter news". *In Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, CA, pp. 331-339.

Likert, R. (1932). A Technique for the Measurement of Attitudes, Archives of Psychology 140, 55.

Maltz, D., Ehrlich, K., (1995). "Pointing the Way: Active Collaborative Filtering". *In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pp. 202-209.

Maslow, A.H. (1962), Toward a Psychology of Being, Van Nostrand, Princeton, NJ

Mooney, R.J., Roy, L., (2000). "Content-Based Book Recommending Using Learning for Text Categorization". *In Proceedings of the 5th ACM Conference on Digital Libraries*, pp. 195-204.

Paiva, A., Machado, I. (1998). Vincent, an Autonomous Pedagogical Agent for On-the-Job Training. In B.R. Goettl, H.M. Halff, C.L. Redfield, & V.J. Shute (Eds.) Intelligent Tutoring Systems, Lecture Notes in Computer Science 1452 (pp. 584-593). New York: Springer-Verlag.

Pazzani, M., (1999). "A Framework for Collaborative, Content-Based and Demographic Filtering". Artificial Intelligence Review 13(5-6), pp. 393-408

Picard, R.W., (1995). "Affective Computing", in M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 321.

Plutchik, R. (1980), Emotion: A Psychoevoluntionary Synthesis, New York: Harper and Row

Reigeluth, C.M. (1996): A new paradigm of ISD? Educational Technology, 36(3).

Razaki, R., Blanchard, E., Frasson, C. (2006). "On the Definition and Management of Cultural Group of e-Learners". *Proceedings of the International Conference on Intelligent Tutoring System: ITS2006*, Jhongli, Taiwan. June 2006.

Razek, M.A., Frasson, C., Kaltenbach, M. (2004). "Building an effective groupware system". *IEEE/ITCC 2004 International Conference on Information Technology*, Las Vegas, NV, USA, April 2004.

Reeves, T., Reeves, P. (1997) Effective dimensions of interactive learning on the World Wide Web in Khan B (ed) Web-based Instruction Educational Technology Publications, Englewood Cliffs, NJ, 59-66.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". *In Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, pp. 175-186.

Ross, P., Jones, J., Millington, M. (1987). "User modelling in intelligent teaching and tutoring". In Trends in Computer Assisted Education. *Proceedings of 6th Lancaster Conference on Computers in Higher Education.* Blackwells: .

Russell, T. L. (1998): The "no significant difference" phenomenon. North Carolina State University, Raleigh.

Saleman, A., Aïmeur, E., Mani Onana, F. S., "Localisation et recommandation d'experts pour le développement d'une aide collaborative", chapter 6, in Développement, intégration et évaluation des technologies de formation et d'apprentissage, S. Pierre (Ed.), Presses Internationales Polytechnique, Montréal, Novembre 2005.

Salton, G., McGill, M., (1983). Introduction to modern information retrieval. New York: McGraw-Hill.

Scollon, C.N., Diener, E., Oishi, S., Biswas-Diener, R. (2004). Emotions across cultures and methods. Journal of crosscultural psychology, 35(3), pp. 304-326.

Self, J.A. 1988. "Bypassing the intractable problem of student modeling". *In Proceedings of the first international conference on Intelligent Tutoring System ITS-88*, p. 18-24, Montréal.

Shardanand, U., Maes, P., (1995). "Social Information Filtering: Algorithms for Automating 'Word of Mouth'". *In: CHI '95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 210-217.

Simmel, G. (1971) On Individuality and Social Forms: Selected Writings (ed. and introd. by D.N. Levine). Chicago, IL: University of Chicago Press .

Sternberg, R. (1997). Thinking styles. New York: Cambridge University Press.

Streeter, L.A., Lochbaum, K.E., 1988 "Who Knows: A System Based on Automatic Representation of Semantic Structure". RIAO '88, 380-388.

Tecuci, G., Keeling, H. (1999). Developing Intelligent Educational Agents with Disciple. International Journal of Artificial Intelligence in Education, 10, 221-237.

Urry, John. (2002). Mobility and Proximity. Sociology, 36, Number 2, 255-274.

Vassileva, J. (1990) A Classification and Synthesis of Student Modelling Techniques in Intelligent Computer Assisted Instruction. in D.Norrie & H.-W.Six (Eds.) Computer Assisted Learning, Lecture Notes in Computer Science No.438, pp.202-213. Springer Verlag.

Vassileva, J. (1998). Goal-Based Autonomous Social Agents: Supporting Adaptation and Teaching in a Distributed Environment. In B.R. Goettl, H.M. Halff, C.L. Redfield, & V.J. Shute (Eds.) Intelligent Tutoring Systems, Lecture Notes in Computer Science 1452 (pp. 564-573). New York: Springer- Verlag.

Victoria, L., Semion, A., Svetlana, B. (1999). "Impact of cross-cultural differences in distance teaching". Paper presented at the 19th World Conference of the International Council for Distance Education, Vienna.

Vivacqua, A., Lieberman H., (2000). "Agents to Assist in Finding Help", In Proceedings of ACM Conference on Human Factors and Computing Systems (CHI-2000), pages 65-72, The Hague, Netherlands.

Weber, G. (1996). Episodic learner modeling. Cognitive Science 20:195-236.

Weber, G., Specht, M. (1997). User Modelling and Adaptive Navigation Support in WWW-based Tutoring Systems. In Jameson, A., Paris, C., and Tasso, C., editors, *User Modelling: Proceedings of the Sixth International Conference*. Wien, New York: Springer-Verlag, 289-300. http://www.um.org/um_97/gz/WeberG.ps.gz

Weber, M., de Melo, F.R., Meireles, V., Nalini, L.E.G. (2004). A Novel Hybrid Intelligent Tutoring System and Its Use of Psychological Profiles and Learning Styles. Intelligent Tutoring Systems 2004: 830-832

Wenger, E. (1987). Artificial Intelligence and Tutoring Systems. Los Altos, CA: Morgan Kaufmann.

Williams, R. (1958) Culture and Society, 1780 – 1950. Penguin, Harmondsworth

Winn, W. (1989): Toward a rational and theoretical basis for educational technology. Educational Technology Research & Development, 37.

Woolf, B., Regian, W. (2000): Knowledge-based training systems and the engineering of Instruction. Handbook of Training and Retraining.

Yin, J., El-Nasr, S., Yang, L., Yen, J. (1998). Incorporating Personality into a Multi-Agent Intelligent System for Training Teachers, In B.R. Goettl, H.M. Halff, C.L. Redfield,

& V.J. Shute (Eds.) Intelligent Tutoring Systems, Lecture Notes in Computer Science 1452 (pp. 484-493). New York: Springer-Verlag.

Appendix A: Modules and functions

A.1 Modules functions

Browser:

Authentication – The users can sign up or sign into the system.

Input: username (String), password (String)

Output: username (String), password (String)

Target: JSP Server

GeneralPageRequest – The user can request a page by clicking on a link

Input: pageName (String), link (String)

Output: pageName (String), link (String)

Target: JSP Server

Display – A webpage is displayed on the interface.

Input: code (HTML)

Origin: JSP Server

Template Converter:

ConvertFile: - Converts a CulturalTemplate to JSP code

Input: CulturalTemplateAddress (String), userSessionAttributes (HTTPSessionAttributes[ ])

Output: code (JSP)

Target: Web Interface

Proxy:

LoginAgentRequest – forwards authentication information to Allocator.

Input: username (String), password (String)

Output: username (String), password (String)

Target: Allocator

SignupAgentRequest – forwards signup information to Allocator.

Input: username (String), password (String), placeOfOrigin (String)

Output: username (String), password (String), placeOfOrigin (String)

Target: Allocator

PageBuilderRequest – forwards pageName to ClientAgent

Input: pageName (String)

Output: pageName (String)

Target: ClientAgent

FirstPageRequest – sends name of first page to client when a new session is created

Input: authenticationVerified (Boolean)

Output: firstPageName (String)

Target: ClientAgent

TemplateConversionRequest – forwards session variables to TemplateConvertor

Input: username (HTMLSessionAttribute)

Output: firstPageAddress (String), username (HTTPSessionAttribute[ ])

Target: TemplateConvertor

SessionAttributeCreation – builds session attributes from notions and resources

Input: notionResourceVector (HashTable[ ])

Output: notionResourceAttributes (HTTPSessionAttribute[ ])

Target: Template Converter

RankResourceRequest – forward resource rank update to Culture Modeler

Input: username(String), resourceName (String), resourceRank (Double)

Output: username(String), resourceName (String), resourceRank (Double)

Target: Culture Modeler

UpdateResourceSeenRequest – forward test question answers to Client Agent

Input: username(String), userAnswers (HashTable)

Output: userAnswers(HashTable)

Target: Client Agent

Allocator:

LoginUserPerform – create an instance of ClientAgent in the client Jade Server

Input: username (String)

Output: clientAgent (Client)

Target: Client Jade Server

FindSolverPerform – asks the OtherClients for preferred Resource for a Notion.

Input: notionsList (String [ ])

Output: notionsList (String [ ])

Target: Client Agent

SelectBestNRV – select the best NotionResourceVector according to similarity.

Input: notionsResourceVectors (HashTable[ ]), similarities (double[ ])

Output: notionResourceVector (HashTable)

Target: Client Agent

SignupToSleepy – announce the creation of a new agent to a Sleepy with a free slot.

Input: userName (String),

Output: UserName (String)

Target: Sleepy Agent

CreateNewAgent – create an instance of a new Client Agent

Input: n/a

Output: client (Client Agent)

Target: Client Agent

PutSleepyAID – setup ClientAgent.

Input: userName String

Output: SleepyagentID (AID)

Target: Client Agent

Client:

CreateSession – create an HTTP session when a user logs in

Input: authenticationVerified (Boolean)

Output: userSession (HTTPSession)

Target: Proxy

ReplyFirstPage – reply the address of this client's first page

Input: firstPageName (String)

Output: username (HTTPSessionAttribute)

Target: Proxy

ListNotionRequest – Asks CultureModeler for All notions present in a cultural page

Input: culturalPageName (String)

Output: culturalPageName (String)

Target: Culture Modeler

LoadClientModelRequest – asks the Sleepy Agent for the appropriate user model.

Input: authenticationVerified (Boolean)

Output: username (String), modelType (String)

Target: Sleepy Agent

FindSolverRequest – asks the Allocator Agent for solvers for notions.

Input: notionsList (String [ ])

Output: notionsList (HashTable)

Target: Culture Modeler

SolveNotionPerform – replies best Resource for appropriate Notion and degree of similarity with the received notionList.

Input: notionsList (HashTable)

Output: notionsResourceVector (HashTable), similarity (double)

Target: Allocator Agent

SolveResource – forwards the preferred resource for a notion to Proxy

Input: notionResourceVector (HashTable)

Output: notionResourceVector (HashTable)

Target: Proxy

ChangeClientModelPerform – update client model with new values

Input: notionResourceVector (HashTable)

Output: modelUpdated (Boolean)

Target: Culture Modeler

ResourcePerform – returns a list of all resources seen in this lesson

Input: lessonName (String)

Output: NotionResourceVector (Hashtable)

Target: Proxy

Sleepy:

SolveGroupNotionPerform – replies best Resource for appropriate Notion and degree of similarity with the received notionList at a group level for all "un-logged" agents.

Input: notionsList (HashTable)

Output: notionsResourceVector (HashTable), similarity (double)

Target: Allocator Agent

LoadClientModelPerform – Load the Client Agent with its model at log-in

Input: username (String [ ])

Output: notionsResourceVector (HashTable)

Target: Client Agent

AllocateSleepyPerform – Send the new user Model to Sleepy Agent at sign-up

Input: username (String), notionResourceVector (HashTable)

Output: clientAllocated (Boolean)

Target: Allocator

Culture Modeler:

SLEEPY*LoadModelPerform* – get the appropriate user model from DB.

Input: username (String), modelType (String)

Output: userModel (HashTable)

Target: Client Agent

ListNotionPerform – gets all notion present in a page from DB

Input: pageName (String)

Output: notionsList (String [ ])

Target: Client Agent

CreateUserPerform – forward user creation request to DB

Input: userName (String), password (String), placeOfOrigin (String)

Output: userName (String), password (String), placeOfOrigin (String)

Target: DB

RankResourcePerform – forward resource rank update request to DB

Input: resourceName (String), resourceRank (Double)

Output: resourceName (String), resourceRank (Double)

Target: DB

ChangeClientModelRequest – request update of client Model upon change in DB

Input: resourceChange (Boolean), resName (String), resRank(Double)

Output: resourceName (String), resourceRank (Double)

Target: Client Agent

EvaluatePerform – forward evaluation of user's answers to final test

Input: userAnswers (Hashtable)

Output: userAnswers (Hashtable)

Target: DB

DB:

UserExists – Checks whether a Client exists.

Input: username (String), password (String)

Output: authenticationVerified (Boolean)

Target: Self

getNRV – get a Client model (NRV) from all NRV.

Input: username (String), modelType (String)

Output: NotionResourceVector (HashTable)

Target: Self

getPageNotions – get the notions present on a page.

Input: pageName (String), modelType (String)

Output: notions (String [ ])

Target: Self

Appendix B: Experimentation and results

B.1 Multiple Choice Question Quiz

Question 1:

What rank does WHO place low fruit and vegetable intake among its 20 risk factors for global human mortality?

Answers:

2nd

6th

20th

Question 2:

Over the past half century, diet preferences have changed to vegetable oils, sugar and meat from?

Answers:

High calory diet

Fruits and vegetables

Cereals and pulses

Question 3:

The Global Fruit and Vegetables Initiative for Health (or GlobFaV) seeks to maximize synergies between WHO's global work on diet, physical activity and health, and FAO's programmes on nutrition, food security and the horticultural supply chain?

Answers:

True

False

Question 4:

Surveys in the US have found that the main barriers to eating more fruit and vegetables are?

Answers:

High cost and poor quality

Low nutritional value

Quick perishability

Question 5:

Micro gardens can be found in?

Answers:

Urban areas

Peri-urban areas

Both

Question 6:

What do farmers need in order to achieve efficient gains?

Answers:

Higher supplies

Access to technology

Both

B.2 Users' results to evaluation

user	number of matches	percentage of matches	number of right answers	grade
1	2	25	7	87,5
2	3	37,5	7	87,5
3	2	25	6	75
4	2	25	6	75

5	1	12,5	7	87,5
6	3	37,5	6	75
7	2	25	5	62,5
8	2	25	7	87,5
9	3	37,5	7	87,5
10	3	37,5	6	75
11	2	25	7	87,5
12	1	12,5	6	75
13	0	0	5	62,5
14	1	12,5	7	87,5
15	2	25	6	75
16	1	12,5	8	100
17	2	25	5	62,5
18	3	37,5	7	87,5
19	3	37,5	7	87,5
20	4	50	7	87,5
21	2	25	6	75
22	1	12,5	6	75
23	3	37,5	5	62,5
24	3	37,5	5	62,5
25	2	25	6	75
26	4	50	6	75
27	3	37,5	7	87,5
28	2	25	7	87,5
29	1	12,5	7	87,5
30	2	25	5	62,5
31	3	37,5	6	75
32	5	62,5	6	75
33	2	25	6	75
34	2	25	7	87,5
35	2	25	6	75
36	3	37,5	6	75
37	4	50	7	87,5
38	4	50	8	100
39	4	50	7	87,5
40	3	37,5	6	75
41	1	12,5	6	75
42	2	25	7	87,5
43	5	62,5	7	87,5
44	6	75	6	75
45	5	62,5	7	87,5
46	4	50	6	75
47	6	75	8	100
48	3	37,5	7	87,5
49	5	62,5	6	75
50	4	50	8	100
51	3	37,5	8	100

X

52	4	50	8	100
53	6	75	7	87,5
54	5	62,5	6	75
55	6	75	8	100
56	4	50	7	87,5
57	3	37,5	8	100
58	5	62,5	7	87,5
59	7	87,5	8	100
60	3	37,5	8	100
61	6	75	7	87,5
62	6	75	8	100
63	5	62,5	6	75
64	7	87,5	7	87,5
65	6	75	4	50
66	5	62,5	7	87,5
67	6	75	6	75
68	4	50	8	100
69	5	62,5	6	75
70	5	62,5	8	100
71	7	87,5	7	87,5
72	4	50	6	75
73	6	75	8	100
74	5	62,5	7	87,5
75	6	75	6	75
76	5	62,5	7	87,5
77	3	37,5	8	100
78	5	62,5	6	75
79	6	75	8	100
80	6	75	7	87,5
81	3	37,5	8	100
82	2	25	7	87,5
83	5	62,5	6	75
84	6	75	7	87,5
85	7	87,5	8	100
86	6	75	7	87,5
87	5	62,5	6	75
88	4	50	7	87,5
89	6	75	8	100
90	5	62,5	7	87,5
91	6	75	8	100
92	4	50	7	87,5
93	6	75	7	87,5
94	5	62,5	7	87,5
95	7	87,5	8	100
96	7	87,5	6	75
97	6	75	6	75
98	4	50	6	75

99	5	62,5	6	75
100	4	50	8	100
101	5	62,5	7	87,5
102	6	75	6	75
103	4	50	7	87,5
104	7	87,5	7	87,5
105	5	62,5	8	100
106	6	75	7	87,5
107	5	62,5	7	87,5
108	6	75	7	87,5
109	5	62,5	7	87,5
110	6	75	6	75
111	5	62,5	8	100
112	5	62,5	8	100
113	6	75	7	87,5
114	7	87,5	6	75
115	6	75	8	100
116	5	62,5	7	87,5
117	6	75	8	100

Appendix C: Computational methods

C.1 tf-idf

From Wikipedia, the free encyclopedia

The tf–idf weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

The *term frequency* in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term ti within the particular document.

$$tf_i = \frac{n_i}{\sum_k n_k}$$

where n_i is the number of occurrences of the considered term, and the denominator is the number of occurrences of all terms.

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$idf_i = \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

With

- $|D|$: total number of documents in the corpus
- $|\{d : d \ni t_i\}|$: number of documents where the term t_i appears (that is $n_i \neq 0$).

Then

$$tfidf = tf \cdot idf$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tends to filter out common terms.