International Journal of Scientific Engineering Research, Vol. 8, No. 1, January 2017
ISSN 2229-5518

A dynamic and hybrid modeling user profiles for custom search

Yannick U. Tchantchou Samen, Eugène C. Ezin

Abstract—During this last decade, many works have been done in order to modelize user interests during his interaction with an information retrieval system and particularly on the web. Despite all these efforts, it remains a real challenge to propose a model able not only to learn user's interests implicitly but also by allowing any user to participate at any moment to build his profile. In this paper, we propose a dynamic and hybrid model able to overcome the aforementioned problem. By using tools as the multi-agent systems, we propose an approach able to collect user interests both explicitly by allowing him to insert or delete his interests in his explicit profile and implicitly while browsing the web. This model takes into consideration the evolutive nature of user's interests and enables us to build an ontological user profile dynamically. The experimentation of our model shows that it is able not only to detect user's interests with an high precision but also to detect changes or drift in these interests.

Index Terms—User profile, reference ontology, Multi-agent system, Hybrid, Dynamic.

1 Introduction

Big data coupled with the recent advances in information and communication technology bring new challenges in the fields of information retrieval. Now, the problem is not about the availability of information but in the ability of the information retrieval system to select and offer the right information that meets the user's needs.
The implementation of search engines does not solve this problem; on the contrary, they merely offer a multitude of answers to the users who feel compelled to search the right information among them. Futhermore, most of these search engines do not take into account the user who is supposed to be the main element of the information search process. All this justify the need to develop information systems that can personalize the information search according to specific user's needs. To get there, it is important to model and build user profile for collecting and detecting its preferences and interests.
The two approaches used to collect user's interests are the implicit approach and the explicit approach. In the explicit approach, user manually creates his profile and enriches it by inserting information about its preferences and interests Ilic et al. [1]; Pannu et al. [2]. Unlike in the implicit approach, user does not intervene directly in the profile creation process; the system based on observed search history and browsing behaviour, collects information about user and creates his profile Shen et al. [4], White et al. [5].

• *Yannick U. Tchantchou is with the Institute of Mathematics and Physics, The University of Abomey-calavi, Benin. E-mail: samenyannick@gmail.com*
• *Eugène C. Ezin is with the Institute of Mathematics and Physics, The University of Abomey-calavi, Benin. E-mail: eugene.ezin@imsp-uac.org*

These techniques have advantages and limitations. For example, used in isolation the explicit method may be accurate but intrusive while the implicit method on the other hand may be transparent to the user but less focused.
By taking in consideration these aforementioned limitations, Pannu et al. [6] have proposed an hybrid system for collecting user's informations both explicitly and implicitly. However, the main limit of this proposed system is that it cannot detect changes in user's interests; but we know that the user's interests are rarely static.
Thereafter, Hawalah and Fasli [7], [8] proposed a dynamic model of user profile by using the ontology concept and the implicit approach to collect user's interests. This model captures user's interests and also detects changes and drift in this interests. However it faces the cold start problem.
In this paper, we propose a dynamic and hybrid user profiling. This model allows us to gather information about user implicitly and explicitly and to detect changes in these user interests. To achieve this, we use the ontology concept to build the ontological user profile, and a multi-agent system for learning user short-term and long-term interests.
The remainder of this paper is organised as follows: in Section 2, we discuss related works, Section 3 we propose the main architecture of the proposed dynamic and hybrid model, and the next Section gives the experimental set-up and evaluation of our proposed model. This paper ends with a conclusion and the outlook.

2 Related Works

The need of modeling user in his interaction with information retrieval systems is a concern that goes back

many years. In the last decade, many research have been done and some authors proposed some approaches for modeling implicitly or explicitly user profile. Indeed, Liana Razmerita et al. [9] proposed a generic architecture of user modeling based on the ontology and they applied this in the context of knowledge management systems. Lin Li et al. [10] introduced an adaptative scheme to learn the changes in user preferences by using the clicks historic. They proposed an independent model for long term and short term preferences of the user to determine his profile. In the same year, Sieg A. et al. [11] defined an approach of personalized search consisting to build the user contextual model as ontological profile.
In the same row, M. Daoud et al. [12] in theirs works proposed an improved approach for learning a semantic representation of user's interests by collecting and representing his search history by using the hierarchy of the ontological concepts. Another technique of user modeling have been proposed by Ashish Nanda et al. [13]. They use general ontology of web and a set of collected web pages, able to give a general idea of user's interests. Kotov, A., et al. [14] proposed methods for modeling and analyzing user search behavior that extends over multiple search sessions. They focus on two problems: given a user query, identify all related queries from previous sessions that the user has issued, and given a multi-query task for a user, predict whether the user will return to this task in the future.
Other group of authors like M. Ilic et al. [1], Pannu et al. [2] have been interested for the user profile modeling based on explicit collection of user's preferences. However, just few authors have proposed an approach based on hybrid collection of user interests. One of the famous work did in this sense is the one did by Pannu in his thesis in 2011. She has proposed a mediation system able to collect user's interests manually by user itself and implicitly by the system.
Another group of persons have used technologies as the multi agent systems for modeling user. It is the case of P.H.H. Rangen et al.[15] who defined a multi agent approach allowing to generate dynamically a knowledge profile and user's interests; Hawalah and Fasli [7] who proposed a multi-agent system using ontological user's profile for dynamic user modeling.

3 A Multi-agent system for building our dynamic and hybrid user profile

In this section, we present a dynamic and hybrid user profile that is able to learn and adapt user's interests based on data obtained explicitly from the user and by observing user behaviour and mapped to a reference ontology. The user's profile consists of four layers: session based layer, explicit profile layer, short-term layer and long-term layer. Each layer consists of one or more agents that are responsible for a set of tasks. The use of multi-agent system resides in their ability to address the complex problem by dividing it into sub problems

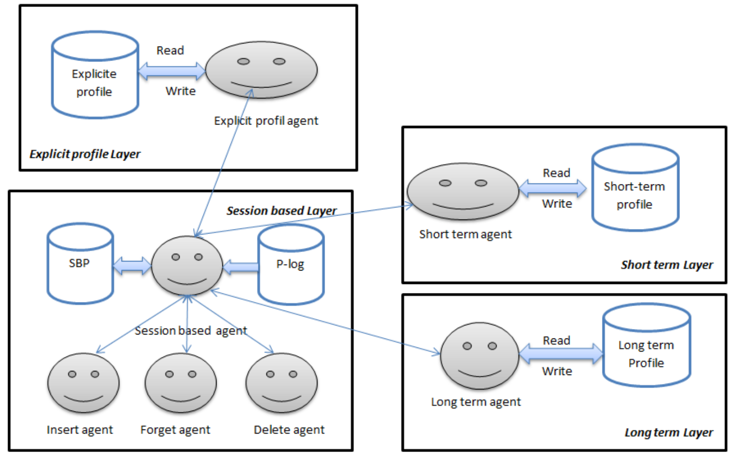

Fig. 1: Architecture of our multi-agent system

which can be handled by agents. The proposed model needs to track user behaviour, add, update, delete user interest and dynamically process explicit user interests. Generally, user interests always change. Indeed, a user may lose an interest in an item or a concept that he was interested in the past. Hence it is important to detect this change in behaviour and to adapt the user profile to improve the user information search. The proposed model is a generalization of the one proposed by Hawalah and Falsi [8] with the capability to learn and adapt to such user behaviour by collecting data explicitly from user and implicitly by observing user browsing behaviour.
We want to offer the possibility to any user to participate when he needs in the process of building his profile. For this purpose our multi-agent system is able to take into account information obtained explicitly from the user and the other one obtained by tracking user behaviour implicitly to detect its interests and therefore any shift and drift in these interests.

3.1 Explicit profile layer

This layer deals with the explicit collection, storage and the updating process of the interests inserted explicitly by the user. Each user input instructions concerns his explicit profile are processed first in this layer by the explicit profile agent and stored in the explicit profile before being sent to the session-based agent for the next step.
A **user explicit interest** is an ontological concept of interest insert explicitly by user in his explicit profile.

3.1.1 Explicit profile agent

This agent is in charge of certain tasks so we can mention among other:

1. collection and storage of information insert explicitly by the user;
2. communication with the session-based agent to advise on all the operations performed by the user

in its explicit profile (insertion of new information, delete, etc.);
3. communication with the user before completing the deletion process of the concept that are still in his explicit profile.

3.2 Session-based layer

This layer has an essential role in the modeling process of our dynamic and hybrid profile. In fact, it contains all the mechanisms associated with learning and adaptation of user interests. For this reason it is related to all other layers in our model.
It receives both explicit concepts inserted by the user and concepts visited by the user during his browsing session and contained in the P-log file (*Processed log file*). Thereafter, it computes and updates the weight of this concept in the session-based profile. Finally it sends the list of this updated concept with their weight to the short-term and long-term layer to determine the short-term interests and the long-term interests. This layer is active during each browsing session to deal with new concepts inserted or visited in order to detect any shift or drift in the user's interests. This layer also includes a profile called SBP (session based-profile) and several agents. Each of these agents is responsible for one or more tasks.

3.2.1 Session-based agent

This agent is the core of our multi-agent system and it is responsible for several tasks:

1. data collection from the P-log file;
2. data collection from the explicit profile;
3. Communication with other agents to calculate the latest interest weight in a session-based profile;
4. Communication with the short-term and long-term agents to enable them to discover short-term and long-term interests respectively.

Once a session ends, data in the P-log file and the other one inserted explicitly by user in the explicit profile are processed and stored in the SBP. Each concept in the SBP has 5 attributes while explicit concept in the explicit profile (EP) has only one attribute (its status). These attributes include:

1) the *status* which can be: positive status such as *browsing concept*, *confirmed concept*, *explicit concept* or *explicit confirmed concept*, or negative Status such as *forgotten concept*, *explicit forgotten concept* and *deleted concept*.
2) the *relevance-size* which refers to how much a concept is relevant to a user interest. As Hawalah and Fasli in [8], the relevance-size can be measured based on user feedback about each concept. If the user feedback is positive, then the relevance-size increases and if the feedback is negative it decreases.
3) the *explicit frecency* which represents the interest's weight of each explicit concept inserted by user in his explicit profile.
4) the *frecency* which represents the interest weight of each concept (explicit concept or not) and it indicates how much a user is interested by this concept.
5) the *frequency* which represents how many URLs from P-log file have been mapped to a particular concept.

The first task of the session-based agent is to extract in the p-log file concepts representing the web page visited by user during the session, their outstanding status, frequency. It also recovers from eventual concept inserted by the user during the same session. Thereafter it communicates with other agents (insert agent, forget agent and delete agent) to determine the relevance-size and compute the frecency of each concept and eventually the explicit frecency. This process is performed daily and all new interests as well as existing ones are processed daily to adapt them to any change in user behaviour.

Algorithm 1: Session-based agent

$P-log = \{\omega_{1-t_s}, \omega_{2-t_s}, \cdots, \omega_{n-t_s}\}$ // P-log holds web pages and their textual contents.
SBP = is the session-based profile that holds user concepts and other attributes.
EP = is the explicit profile that holds user explicit concept.
Input: The last session in the explicit profile, in the P-log file that holds web pages, duration and timestamp, and the session-based profile.
Output: Updated session-based profile.

// process all the new visited web pages in the last session from the P-log file and the new explicit concept in the explicit profile by sending them to the Insert agent.
foreach $\omega_i \in$ *P-log: Last session* **do**
  $New.c = \omega_i.Extract(concept)$;
  $New.c_{duration} = \omega_i.Extract(duration)$;
  Send New.c and New.$c_{duration}$ to insert agent;
end
foreach $c_i \in EP$ *: Last session* **do**
  send c_i and c_i.Status to insert agent;
end
// (2) Process all the existed concepts in SBP that have not been updated.
foreach $c_i \in SBP$ **do**
  if c_i*.Status = forgotten concept or explicit forgotten concept* **then**
    send c_i, c_i.duration and c_i.relevance-size to Forget agent;
  else
    $c_i.Status =$ deleted concept;
    send c_i, c_i.duration to Delete agent;
  end
end

3.2.2 Insert agent

This agent is responsible for processing all concepts with positive status received from the Session-based agent. Unlike Hawalah model [8], we distinguish four different events with positive status: *browsing concept*, *confirmed concept*, *explicit concept* and *explicit confirmed concept*.
The *browsing concept* event is the default one that is assigned to any concept browsed by the user. If such concept appears in two subsequent sessions, it is assigned to *confirmed concept* event. The later event has a higher weight than the *browsing concept*, as concepts appearing in more than one session would likely be of more interest to users than those that appear one.
The *explicit concept* event is any concept inserted explicitly, or manually by the user in its explicit profile. As information given by the user seems much more credible than those implicitly obtained by the system, this event has a higher weight than the *confirmed concept*. If such concept appears implicitly in one or more sessions, it is assigned *explicit confirmed concept* event. The later event has a higher weight than the *explicit concept*, as it is firstly an explicit concept but also confirmed implicitly during the browsing session as a interesting concept. Furthermore, it should be emphasized that any confirmed concept can become an explicit confirmed concept; but the opposite is not possible. For example, the user can explicitly insert concept that is already recognized in the system as a confirmed concept. After insertion, this concept will be assign *explicit confirmed concept* event. Finally, if when the concept is inserted explicitly by the user in its explicit profile, it is not yet included in the session-based profile (or has never been implicitly identified as a concept of interest to the user), then this concept should appear in another browsing session to acquire the status of *explicit confirmed concept*.
In the proposed model, the explicit frecency (ExpFre) value is computed only at the moment when this concept is explicitly inserted by user in the explicit profile. As it is more difficult to determine how many explicit concept is relevant to a user interest, we propose to use weight associated to the status of this concept and the frecency of each concept in the SBP in the previous session. Unlike Pannu et al. [6] who consider the weight of an explicit concept as the highest weight of all the implicit concept in the profile before the insertion of this explicit concept, we use the following equation:

$$ExpFre = \frac{\sum_{c \in SBP_{s_i^*}}^{n} Fre}{N} + E.Weight \tag{1}$$

where N is the total number of concepts in the SBP at this moment, E.Weight is the event weight of this concept and $\sum_{c \in SBP_{s_i^*}}^{n}$ Fre is the total frecency value of all concept c in the SBP before the session s_i.

$$E.Weight = \begin{cases} 100 & \textit{if browsing-concept} \\ 150 & \textit{if confirmed-concept} \\ 200 & \textit{if explicit concept} \\ 250 & \textit{if explicit confirmed concept} \end{cases} \tag{2}$$

We don't want to take the highest weight as Pannu et al. [6], for the reason that the user may make an error on the concept entered as explicit interest. At this point, it would not be appropriate to assign the highest weight while it does not exactly reflect the reality. Moreover E.Weight allows us to quantify the interest of a concept inserted explicitly by user if it uses the proposed system for the first time. As SBP does not contain concept at this time, the explicit weight of this concept will be reduced to E.Weight.
Finally, it is important to say that in our proposed model, the explicit frecency of each concept depends on the time of his insertion in the explicit profile and the frecency of each concept in the SBP at this moment.
Moreover, calculating the frecency of a concept depends on the status of this concept and the duration that is associated with the web page. The frecency of that concept is accumulated using equation 3

$$Fre = \sum_{c_i \in C} \left(\frac{c_i.k}{100} + \alpha\right) * E.Weight + \alpha * Frecency\_average_{s_i^*} \tag{3}$$

where $\alpha = 1$ if this concept is in the explicit profile and $\alpha = 0$ if not. $c_i.k$ is the visit duration of the page c_i, E.Weight is given by the equation 2. After computing the frecency weight of each new concept, the Insert agent sends this concept and its properties to the SBA.

3.2.3 Forget agent

This agent handles the behaviour that occurs when user loses interest in a concept. Our system is able to take into account these changes in user's behaviour. However, this concept cannot be deleted immediately. They must follow a gradual forgotten process to be confirmed as an uninteresting concept for user.
In our case, the forgotten process depends on several factors:

1) the *relevance-size* that is associated with each concept. The relevance-size is an indicator of the user's strength of interest in a concept. The value of the relevance-size is essential to determine the pace of the forgotten process as the larger the relevance-size is, the slower will be the forgotten process and vice versa;
2) the *recency* of a concept, as old concepts are forgotten faster than new ones;
3) the introduction of new interests to a user profile. If a user has started to lose his interest in a concept, and at the same time started to show interest in new concepts, then this behaviour might indicate that a user has started to drift his interest to new concepts;

Algorithm 2: Insert agent

SBP = is the session-based profile that holds user concepts and other attributes.
EP = is the explicit profile that holds user explicit concept.
Input: concept, concept.duration and concept.Event, received from the session-based Agent
Output: Updated session-based profile.

//(1) Discover the event that belongs to the positive status;
c_i.Discover(Event);
if *Event = browsing concept* **then**
  c_i.Event = browsing concept;
else
  if *Event =confirmed concept* **then**
    c_i.Event = confirmed concept;
  end
end
set c_i.relevance-size $+= 1$ //Because the positve status increases the RS by one;
set $c_i.frecency += (\frac{c_i.duration}{100}) * E.Weight$;
$c_i.ExplFre = 0$;
SBP-Add-or-Update (c_i, c_i.ExplFre, c_i.frecency, c_i.relevance-size and c_i.Event)
if this concept is new, then add it to the SBP, otherwise update it.;
if *Event= explicit concept* **then**
  c_i.Event = explicit concept;
else
  if *Event= explicit confirmed concept* **then**
    c_i.Event = explicit confirmed concept;
  end
  set c_i.relevance-size $+= 1$;
end
if *c_i is new in EP* **then**
  set $c_i.ExpFre$ =average-frecency (of concept in the SBP at before this session) +E.Weight;
  set $c_i.frecency+ = (\frac{c_i.duration}{100}) * E.Weight + c_i.ExpFre$;
else
  set $c_i.frecency+ = (\frac{c_i.duration}{100}) * E.Weight$;
end
SBP-Add-or-Update (c_i, c_i.ExplFre, c_i.frecency, c_i.relevance-size and c_i.Event) ;

4) the user decides to explicitly delete this concept in his explicit profile. In this case, the explicit frecency of this concept will be reduced to zero and his frecency will reduce considerably at this time. This factor could help us to detect a drift in the user interets as if at the same time user add new concept in his profile this event will reflect this drift.

To apply our forgotten process, we propose a generalization of the formula proposed by Hawalah and Falsi in [8] by taking into account the eventuality that user explicitly deletes the concept in its explicit profile. We emphasize that this scenario is only possible if the concept was already in the user explicit profile. In this case, the new frecency of the forgotten concept is given by the equation 4.

$$c_i.NewFre = (c_i.OldFre - \alpha ExplFre).e^{-(\frac{log2}{c_i.RS*2}(G_m - G_l) + N.New)} \quad (4)$$

where $\alpha \in \{0,1\}$, with $\alpha = 1$ if user explicitly deletes the concept in the explicit profile and $\alpha = 0$ if not. c_i.RS is the relevance-size of a concept c_i, G_m is today's day, G_l is the day of the last occurrence and $N.New$ is the number of new interests that is introduced in the G_m day.
The fact that the user explicitly removes the explicit concept in its profile does not allow the system to delete immediatly this concept in the SBP. This factor accelerates the forgotten process, which ends when the relevance-size decreases to zero. In this case, the concept will be sent to Delete agent and will be removed from the SBP. Furthermore, it should be emphasized that the forgotten process may be paused when user shows interest again to the concept. Then the relevance-size will increase to reflect this change.

3.2.4 Delete Agent

As Hawalah and Fasli [8], this agent manages the gradual deletion of a concept from a user profile. When a concept is passed on to the Delete agent, this is removed much faster based on the time of the last appearance of the concept, and until the weight reaches a predefined threshold and then it is removed altogether. We also use the following equation to compute the new frecency of a deleted concept:

$$c_i.NewFre = \frac{c_i.OldFre}{G_m - G_l} \quad (5)$$

Where G_m is current date and G_l is the date of the last occurrence of c_i.
It is important to note that prior to trigger the process of removing an inserted concept by the user and that is always present in his explicit profile, the system must firstly send a message to the user to inform the forthcoming abolition of the concept. If the user explicitly

Algorithm 3: Forget agent

SBP = is the session-based profile that holds user's concepts and other attributes.
EP = is the explicit profile that holds user explicit concept.
Input: concept, concept.duration and concept.status, concept.frecency and concept.relevance-size received from the Session-based agent
Output: Updated session-based profile.

if c_i.*relevance-size* > 0 **then**
  if c_i.*Status* = *explicit forgotten concept* **then**
    set $c_i.NewFre = (c_i.OldFre - ExplFre)e^{-(\frac{log2}{c_i.RS*2}(G_m - G_l)+N.New)}$;
  else
    if c_i.*Status* = *forgotten concept* **then**
      set $c_i.NewFre = c_i.OldFre.e^{-(\frac{log2}{c_i.RS*2}(G_m - G_l)+N.New)}$;
    end
  end
  set c_i.relevance-size $--$;
end
SBP Update (c_i, c_i.frecency, c_i.relevance-size and c_i.status).

confirms he is still interested in the concept, then it will be considered a new explicit concept inserted by the user at that time and explicit frecency and frecency will be recalculated. Such a concept will reassign a positive status and will be treated as such. Otherwise the removal process should trigger until the final deletion of this concept.
After having processed the whole concepts, the session based agent stores the final results in the session based profile. At this time, all the concepts are ready to be use to discover the short-term and the long-term interests. So the last task of the Session based agent is communicating with short-term and long-term agents to help them discover these interests.

Algorithm 4: Delete agent

SBP = is the session-based profile that holds user's concepts and other attributes.
EP = is the explicit profile that holds user explicit concept.
Input: concept, concept.duration and concept.status, concept.frecency and concept.relevance-size received from the Session-based agent
Output: Updated session-based profile.

if c_i.*relevance-size* $= 0$ **then**
  set c_i.status = deleted concept;
  set $c_i.NewFre = \frac{c_i.OldFre}{G_m - G_l}$;
  if $c_i.frecency \leq$ *threshold* **then**
    SBP-delete(c_i);
  else
    SBP Update (c_i, c_i.frecency, c_i.relevance-size and c_i.status);
  end
end

Short-term and long-term interests are paramount in personalization systems in that they promote more efficient personalization of information by taking into account these user short-term and long-term preferences. However, the deletion process of this interest remains complex and also specific to the type of desired interest (short-term or long-term). Continuing the Hawalah work, we are redefining the role of short-term and long-term layers and their respective agent, by taking into account the possibility of an explicit inclusion of concepts by user.

3.3 Short-term layer

The short-term layer is in charge of the development of learning mechanism of user's short-term interests. This layer includes the short-term profile (STP) used for storing all the concepts identified as short-term interests, and the short-term agent (STA) that is responsible for tasks such as discovering, maintaining and storing short-term interests in the short-term profile.
As in Hawalah and Falsi [8], the detection process remains the same. However, based on the frecency value of each concept in the user SBP, we calculate a threshold so each concept with frecency value above it will be considered as a short-term interest. Note that this threshold is not fixed, his value depends on each user's browsing behaviour and the frecency value of each concept in the SBP. Instead consider our threshold as the ratio of the total frecency values for each visited concept in each session by the number of concepts that have been visited in all the sessions. Our threshold is computed as the ratio of the total frecency values of each concept in the SBP to the total number of concepts in SBP at this moment.

$$Threshold = \frac{\sum_{c \in SBP} Frecency\_values}{|Total \quad number \quad of \quad concepts|} \tag{6}$$

Let us note that this formula is more general than the one proposed by Hawalah [8]. Indeed it takes into account the eventuality where the SBP contains explicit concept i.e concept inserted explicitly by the user and which have not still been browsed during a sessions.
Hence, all the concepts in the SBP that have weight above this threshold are stored in the STP. As this threshold adapts to various browsing behaviours, each user would have a different threshold.
In our model, an explicit concept is not necessarily a short-term interest; unless its frecency is greater than the threshold. Indeed the frecency of each explicit concept is equal to the explicit frecency of this concept as it is the first time that it appears in the SBP. However, the explicit

frecency obtained by equation 1 does not guarantee that it will be greater than the threshold obtained after the current session. This eventuality depends on the user browsing behaviour during this session.

3.4 Long-term layer

This layer learns, recognizes and stores the user's long-term interests. It also includes two components: the long-term profile (LTP) that stores all interests that are recognized as long-term ones, and the long-term agent (LTA) whose task is to recognize , maintain and store long-term interests in the long-term profile.
In our model, any explicit concept cannot be consider as a long-term interest. The reason is that user has just insered this concept as an explicit concept but not ever browsed it during one of his browsing session. In fact, to calculate the frequency weight for each concept in the SBP, the LTA use the following equation:

$$c_i.FW = (N_{occ} * N_{day}) * (\frac{G_m - G_f}{d_0}) \tag{7}$$

where N_{occ} is the number of times the concept c_i occurs in our system, N_{day} is the number of days that concept c_i occurs in, G_m is the day when the process is launched, G_f is the day of the first appearance of the concept c_i and finally, the d_0 is the maximum of all the N_{occ}. Let us note that in this equation, the priority is given to concept having spent more time in our system. Otherwise, this formula is more dynamic than the other proposed by Hawalah which depends on the age of the P-log file.
In order to identify the long-term interests, we have proposed an extension of the Hawalah's proposed threshold. The feature of our threshold lies in the fact that it depends on the status of the concept. As user can insert concept explicitly in his explicit profile, we believe that such a possibility should be considered in the process of detection of long term interests. For this purpose, we determine this threshold using the following equation.

$$Threshold = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(c_i.FW - AV)^2 + \frac{\sum_{i=1}^{N} c_i.FW}{\xi \cdot |N|}} \tag{8}$$

where N is the total number of concepts, $c_i.FW$ is the frequency weight of a concept c_i, AV is the average of frequency value of all the concepts and $\sum_{i=1}^{N} c_i.FW$ is the total of all the frequency value. $\xi \in \{1, 2\}$, where $\xi = 2$ if the concept was inserted explicitly by user and $\xi = 1$ if not.

4 Experimental set-up and evaluation

The evaluation process of user profile model remains a difficult task because of the complexity of elements to take into account. In our case, we need first of all to collect and process user's data, and to use this processed data to detect the user interests. For this purpose, we built a web browser and used to collect user browsing behaviour (visit web page, duration, content of each web page and timestamp). This web browser also has an interface for explicit insertion and deletion of user's interests. We did this evaluation in a real environment with a real users.

4.1 Information retrieval phase

To use our evaluation framework, each user must possess an account if it is the first time that he uses it and then he must identify himself using his login and password. During the registration in our framework, the user may insert his interests and other information. This inserted interests are directly collected by the explicit agent and treated by the multi-agent systems.
When the user browses the web, our framework collects implicitly all the visited web pages, duration of visit and stores in the log file database. When the user leaves this web page, our framework extracts this content, processes and maps it to the corresponding ontological concept. Let us note that in our case, we use a reference ontology "Computer science" of *open directory project* (ODP) [1].
The content of each visited web page is processed as follows: firstly we remove all the stop words by using the Porter algorithm (1997) [16], secondly by stemming process we reduce each word to his stem. By going in the same direction as Pannu M.[6] and White [5], in our experimental framework, the weight of each word in the web page depends on two components. The word position in the web page and the number of occurrences of this word at this position. Indeed, a word located in the title of a web page represents more the content of this page than another word located in the body of this page. For this reason, we first of all define a weight corresponding to the position of the word in the web page. So, for a word located in the title, we attribute a weight of 0.5 while for a word in the metadata we attribute a weight of 0.3 and finally a word in the body a weight of 0.2. The final weight of the word for this page is given by the following formula.

$$\omega_{t_i} = \sum_{j=1}^{3} \alpha_{t_i,j} p_{t_i,j} \tag{9}$$

where $p_{t_i,j}$ is the weight of the word t_i in the position j, $\alpha_{t_i,j}$ is the number of occurrences of the words t_i in the position j and finally j represents title, metadata or the body of the web page.
This formula allows us to determine the term vector representing this page with the weight associated to each of this word.
After this, we use the traditional cosine similarity [17] to map the visited web pages to the ontological concept

1. http://www.dmoz.org/

from the reference ontology.

$$Sim_{cosine}(d_1, d_2) = \frac{\sum_{i=1}^{n} \omega_{i_1} \cdot \omega_{i_2}}{\sqrt{\sum_{i=1}^{n} \omega_{i_1}^2} \cdot \sqrt{\sum_{i=1}^{n} \omega_{i_2}^2}} \quad (10)$$

At the end of this process, the ontological concept with the greatest value of cosine similarity is mapped to this page and is stored into the database of processed log file (Plog).
When the session ends, concepts stored in the database during the session is extracted by the session based agent and treated by our proposed multi-agent system.

4.2 Description of experimental phase

To assess the accuracy of the collection and detection of user's interests, we experimented our model by simulating three scenarios during the period of fifteen days.
In the first scenario, the user during the registration in our framework, inserts three concepts of interests. But during the first five days, he browses only two of these concepts. After this period, he inserts a new concept and the eighth day, the concept one of the concepts in the explicit profile is removed and replaced by a new concept. During the rest of the time, the user browses each of the concept in his explicit profile with the exception of one of the inserted concept at the beginning of the experimentation.
At the beginning of the second scenario, user inserts two concepts in the explicit profile. But during the first three days he doesn't browse any of these concepts. In the sixth day, he inserts a new concept (the user is not really interested by this particular concept) and the next day this concept is explicitly removed by user. Between the eighth and the eleventh day, he inserts the new concepts and thereafter he browses these concepts and those in the explicit profile.
In the last scenario, the user does not insert concept at the beginning of the process. He browses some concepts in the web without insert it in the explicit profile. The fourth, fifth and seventh day, He inserts respectively a new concept in his explicit profile. The tenth day he removes one of this concept and replaces by a new concept and browses each of the concepts in his explicit profile during the other days.

4.3 Evaluation of the accuracy of our proposed model

At the end of the experimentation phase, a total of web pages have been browsed. We evaluted the accuracy of our information retrieval approach and computed the precision for the user's profiles in term of capturing and adapting to user's interests by using the next formula.

$$Precision = \frac{|Correct\ captured\ and\ learned\ interests|}{|Number\ of\ interests\ recorded\ by\ a\ user|}. \quad (11)$$

TABLE 1: The overall performances of our model

	Accuracy of our IR process	Accuracy of learning process
Scenario 1	0.934	0.78
Scenario 2	0.91	0.86
Scenario 3	0.938	0.95

The results are contained in the Table 1.
Overall we can note that the accuracy of our model in terms of detection of user's interests is very high. The low value obtained in the first scenario is due to the fact that our model does not use the *Replacement-task* proposed in Hawalah [8]. So many of the new browsing concepts are not detected as short term interest; as his frecency is smaller than the threshold. Otherwise some visited web page have been mapped to a parent concept in our reference ontology; as this concept is too general than the corresponding concept.
We have confirmed during the first three days of the last scenario that when user doesn't insert explicitly any concept in his explicit profile, our model behaves exactly as Hawalah proposed model. Futhermore, we have noted in the second scenario that when a wrong concept is inserted explicitly by user, and thereafter he removed after a short period, this concept is directly deleted in the session based profile.
Finally, we can point out that the detection process of long term interest is very selective as the threshold is very high. However, we noticed after this 15 days that our proposed threshold have allowed to detect long term interest in the first and the third scenario while the Hawalah's proposed threshold have not detected any long term interest. This is due to the fact that our approach take into account the status of each concept in the session based profile. All the detected long term interests are concepts that have been inserted explicitly by the user and browsed during the user browsing session.

5 Conclusion

We have proposed in this paper a dynamic and hybrid user profile based on ontology and able to collect the user's data in hybrid way (implicitly and explicitly). The experimentation of our model shows that it is able to process dynamically all the information collected in our system. Futhermore, this model improves the Pannu's proposed model [6] in fact that it takes into account the change in the user's interets. Otherwise, our model is a generalization of the one proposed by Hawalah and Fasli [8] in that it is able to collect and process user data explicitly.
In the future, it will be interesting to use our model to personalize a user's search results on the web. As we do not take into account the semantic structure of our ontological concept, it will be very interesting to study this aspect.

References

[1] Ilic, M., Leite, J., and Slota, M. Explicit Dynamic User Profiles for a Collaborative Filtering Recommender System. IBERAMIA, LNAC5290, pp: 252-261, springer, Heidelberg (2008).

[2] Pannu, M., Anane, R., James, A. Explicit User Profiles in Web Search Personalization; computer supported cooperative work in design (CSCWD), 2011 15th international conference on, 8-10 june 2011. pp. 416-421.

[3] Srinvas, S. Explicit User Profiles for Semantic Web Search using XML. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, vol. 2, Issue 6, November-December 2012, pp. 234-241.

[4] Shen, X., Tom, B., Zhou, C. Implicit User Modelling for Personalized Search. Proceeding of the 14th ACM international conference on information and knowledge management (New York) pp: 824-831.(2005).

[5] White, R. W., Jose, J. M., and Ruthven, I. An Approach for Implicitly Detecting Information Needs. Proceedings of the twelfth international conference on Information and knowledge management (CIKM'03), 2003, pp.504-507.

[6] Pannu, M., Anane, R., James, A. Hybrid Profiling in Information Retrieval; computer supported cooperative work in Design(CSCWD), 2013 IEEE 17^{th} international conference on computer supported cooperative work in design.

[7] Hawalah, A., and Fasli, M. (2011). A Multi-agent System using Ontological User Profiles for Dynamic User Modelling. InProceedings of the 2011 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology (pp.430-437).

[8] Hawalah, A., and Fasli, M. (2015). Dynamic User Profiles for Web Personalization. Journal of Expert systems with Applications, 42, pp. 2547-2569. Doi: 10.1016/j.eswa.2014.10.032.

[9] Razmerita, L., Angehrn, A., Maedche, A. Ontology-based User Modeling for Knowledge Management Systems. In: International Conference on User Modeling, pp. 213 217. Pittsburgh, PA (2003) Resource Description Framework, http://www.w3c.org/RDF/.

[10] Li, L., Yang, Z., Wang, B., and Kitsuregawa., M. Dynamic Adaptation Strategies for Long-term and Short-term user profiles to personalize search. (2007), springer-verlag Berlin Heidelberg. pp. 228-240.

[11] Sieg, A., Mobasher, B., Burke, R., 2007. Ontological User Profiles for Representing Context in Web Search, in: 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, IEEE. pp. 91 94.

[12] Daoud, M., Tamine-Lechani, L., Boughanem, M., 2008. Learning Implicit User Interest using Ontology and Search History for Personnalization. Web information systems engineering-WISE 2007 workshops, vol. 4832, pp. 325-336.

[13] Ashish Nanda, Rohit Omanwar Bharat Deshpande (2014): Implicitly Learning a User Interest Profile for Personalization of Web Search Using Collaborative Filtering. In: Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02, WI-IAT 14, IEEE Computer Society, Washington, DC, USA, pp. 5462, doi:10.1109/WI-IAT.2014.80.

[14] Kotov, A., Bennett, P., White, R., Dumais, S. and Teevan, J. Modeling and Analysis of Cross-session Search Tasks. SIGIR 11.

[15] Rongen P.H.H., Schroder J., Dignum F.P.M., et al.(2005). Multi Agent Approach to Interest Profiling of Users. Proc. of Multi-Agent Systems and Applications IV, pp. 326 355.

[16] Porter, M.F., 1997. An Algorithm for Suffix Stripping, in: Sparck Jones, K., Willett, P. (Eds.), Readings in Information Retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 313 316.

[17] Baeza-Yates, R. and Berthier Ribeiro-Neto. Modern Information Retrieval. AddisonWesley, 1999.

[18] Godoy, D., Amandi, A. Interest Drifts in User Profiling: A Relevance-based Approach and the Analysis of Scenarios. The computer Journal, 52, $771 - 788$.

[19] Trajkova, J. and Gauch, S.(2004). Improving Ontology-based User Profiles, in: Proceedings of large scale semantic access to content (text, image, video, and sound) RIAO, pp. 380-389.