Distributed Knowledge Networks: Towards a New Paradigm for Delivering Knowledge

Alfredo Cuzzocrea
Department of Electronics, Informatics, and Systems
University of Calabria, Italy
cuzzocrea@deis.unical.it

Fernando Bellas
Department of Information and Communications Technologies
University of A Coruña, Spain
fbellas@udc.es

Abstract

Knowledge personalization is currently the most investigated issue in the context of service-oriented systems. Knowledge representation and management are the subtended critical issues for knowledge personalization, and currently they are largely studied, principally thanks to the explosion of data modelling technologies such as XML and XMLSchema. Despite some progress, currently, a widely approved standard for delivering knowledge is still missing. In this paper we propose the definition of Distributed Knowledge Networks (DKN), an open framework for delivering personalized knowledge. We also provide a reference architecture for DKN networks.

1. Introduction

It is nowadays recognized that the personalization of presentations and contents (i.e., their adaptation to user’s requirements and goals) is becoming a major requirement for Services Providers. Application fields where personalization can be useful are manifold; they comprise on-line advertising, direct web-marketing, electronic commerce, news delivery, on-line learning and teaching etc. The need for adaptation arises from different aspects of the interaction between users and services. For instance, in e-commerce systems, having a wide product warehouse is not enough if proposing the appropriate products based on effective (or probable, at least) user needs is not possible.

Currently, the problem of web personalization is addressed through approaches based on clustering methodologies which group users based on similarities of navigation through a given web system. This traditional approach results in poor adaptation because it does not consider the actual user needs. Instead, it tries to infer them by observing what other users with “similar” behaviour have previously done.

This paper introduces a novel approach whose goal is to increase adaptation accuracy in the context of the Knowledge Management Systems (KMS) [6,9,10,14,15]. Starting from the basic issue of the knowledge representation, our approach proposes to manage knowledge according to a set of user profiles, which are clustered on the basis of the user behaviour monitoring. In order to improve user model accuracy, we propose to adopt OLAP technology [8] to define a multi-dimensional user model that presents a high degree of granularity in managing knowledge personalization. More in detail, our proposal is based on the continuous observation of users, whose behaviour is analysed in order to infer their tastes, preferences and needs. A user profile is built from the user’s repeated browsing behaviour and is no longer based on what other users with similar behaviour have previously done. In order to effectively increase adaptation performance, each user is not treated as an anonymous user, but instead user registration information is taken into account as the major requirements for accessing the system.

The rest of this paper is organized as follows: Section 2 describes related work; in Section 3 we provide an overview about Distributed Knowledge Networks (DKN); Section 4 is devoted to illustrate our proposal about the implementation of a DKN network improved by some ad-hoc software components able to effectively manage the knowledge delivery; core methodology for knowledge personalization is presented in Section 5; finally, in Section 6 we draw our conclusions and future work.

2. Related Work

Because of the complexity of a DKN network, there are three major research areas as background for our work: Knowledge Management Systems; User Behaviour Tracking; User Profile Modelling.

Knowledge Management Systems. Briefly, KMS systems [6,9,10,14,15] are usually provided with web
interface systems that should be able to provide the right information to the right user at the right time and in the right format. Application fields of the KMS systems include: e-learning systems, health-care systems, e-procurement systems, e-government systems, industrial building management systems etc. Enterprise Resource Planning (ERP) [2] represents an effective implementation following the KMS systems deployment paradigm. In some previous papers we have proposed valuable improvements respect to the commonly accepted architecture for the KMS systems. In particular, in [4] we have proposed an innovative reference architecture for Knowledge Management-based Web Systems (KM-bWS), which are web systems that exploit the functionalities of the traditional KMS systems and make them more general and usable in other application fields. In [5] we have proposed an extension of the previous work, adding to the reference architecture useful functionalities for improving the web resources searching and classification phases.

**User Behaviour Tracking.** A well-known class of systems that are based on the user behaviour tracking are the so-called Recommender Systems [11]. These systems learn about user preferences over time and automatically find things of similar interest, thus reducing the burden of creating explicit queries. They dynamically track users as their interests change. However, such systems require an initial learning phase where behaviour information is built up to form a user profile.

Content-based recommender systems recommend items with similar content to things the user has shown interest in before. Examples of content-based recommenders are Fab [1] and ELFI [12]. Another interesting work focused on Knowledge Management is [7]; the author proposes a method for finding people that work in similar areas, by applying clustering and Information Retrieval (IR) technologies.

**User Profile Modelling.** For our work, methodologies and techniques oriented to define user profile models are also relevant. For example, in [13] the authors describe a new scheme to learn dynamic user interests in an automated information filtering and gathering system running on the Internet, using a learning algorithm derived for the particular representation of the user interests. In [3] the authors present a probabilistic model for mapping the user behavior into a weighted digraph and an algorithm that operates on this digraph for dynamically inferring the user profile among a set of given profiles.

3. Distributed Knowledge Networks

In this Section we describe the architecture and the main functionalities of DKN networks. A DKN network (see Figure 1) is obtained by interconnecting different KMS systems, comprising both the wired and the wireless domain, and it is able to provide knowledge to the user independently of his/her current position and his/her current connecting device. Recent developments in the wireless technology make feasible this scenario since current handheld devices are equipped with sufficient capabilities in terms of available memory size and computational power.

A DKN network is intended to support distributed knowledge provision and management. Total knowledge of the system is the result of single contributions of the components belonging to the DKN network, where each component is specialized on a particular aspect of the knowledge domain. A user can extract knowledge from different distributed sources according to his/her profile and can use this knowledge in an integrated “transparent-for-the-user” fashion. Moreover, knowledge delivery is adaptive because the system automatically selects the most useful and suitable information for the user, and subsequently provides this information by applying the most convenient formatting task according to the current device.

In this delineated scenario, a user registers himself on the system setting an initial, generic profile according to the his/her preferences/needs and subsequently he/she accesses the knowledge available on the system by using wired and wireless devices that adhere to the DKN network. During the history of interactions between the user and the system, the user profile can be updated and the user can be moved from a profile to another one. As a consequence, the middleware layer of the system is a state-aware layer. In fact, because of the complexity of the system, its evolution is described and managed using the state system abstraction, based on the State Design pattern in the context of software engineering. So, all the interactions of a certain user \( U_i \) with the system are mapped into a finite state automata \( A(U_i) \), where each node \( Q \) represents a state of the system and each arc \( B \) represents a state transaction. For instance, in the context of the e-learning systems, a state transaction could capture the situation in which a user reaches a new learning level, more advanced than the previous one. When user \( U_i \) interacts with the system, the middleware layer of the system updates accordingly the involved states of the automata \( A(U_i) \) and, eventually, its topological configuration.

Formally, the supported scenario is as follows. We have a set \( \Phi \) of \( N \) Services Providers \( \Phi = \{ SP_0, SP_1, \ldots, SP_{N-1} \} \), a set \( \Theta \) of \( P \) services exported by the Services Providers \( \Theta = \{ S_0, S_1, \ldots, S_P \} \) and a set \( \Gamma \) of users \( \Gamma = \{ U_0, U_1, \ldots, U_M \} \). Each user can use different devices belonging to the set \( \Omega = \{ D_0, D_1, \ldots, D_Q \} \) for accessing one or more of the exported services. Given a user \( U_i \),

\[ \ldots \]
each session \( T_m \) of the system is represented by a tuple \( t_{m,i,A,D} = \langle U_i, \{S_h, S_{h+1}, \ldots, S_{h+k}\}, D \rangle \). The meaning of \( t_{m,i,A,D} \) is the following: the user \( U_i \) accesses, using the device \( D_h \), the services belonging to the set \( A = \{S_h, S_{h+1}, \ldots, S_{h+k}\} \).

![Figure 1. Distributed Knowledge Network](image)

Next we provide a formal theoretical framework for representing and extracting knowledge. First of all, we denote by \( \{N_0, N_1, \ldots, N_{Q-1}\} \) the set of knowledge nodes; each knowledge node \( N_i \) can represent both a wired device or a wireless device, belonging to the set \( \Omega \).

We denote by \( \Sigma(N) \) the knowledge inside the node \( N_i \) that is, the network of inter-related concepts held in the node \( N_i \). For instance, in an e-learning system \( \Sigma(N) \) could represent the set of topics belonging to an e-course provided by a server \( N_i \) in a university. The network of concepts can be stored into XML files, since the well-noted properties of XML effectively support the need of flexibility and frequent updates on the data as required by the context of the knowledge management. For instance, an e-course in Computer Science could be represented as follows:

```xml
<teaching_course>
  <name>Data Base Systems</name>
  <class>Computer Science</class>
  <topics>
    <topic ID="CS_DB_0001">
      <name>Entity Relationship Model</name>
      <location xlink:type="simple" link:href="file://topicsRepository/dataBase/topics/ER.xml"/>
    </topic>
    [...]
  </topics>
  <related_topics>
    <name ID="CS_MD_0001">Domain Modelling</name>
    <name ID="CS_MD_0009">Constraints Management</name>
  </related_topics>
</teaching_course>
```

We denote by \( \Pi(N) \) the knowledge extraction task performed on the node \( N_i \). In contrast with the previous case, this formalization is tightly tied with the specific application domain. Without loss of generality, we can think of this task as the selection of a subset of \( \Sigma(N) \), i.e., \( \Pi(N) = \prod_D \Sigma(N) \), such that \( D \) is the selecting (concept) set and a concept \( C \) belonging to the set \( \Sigma(N) \) is projected into the set \( \Pi(N) \) iff \( C \cap D \neq \emptyset \). The process generating \( \Pi(N) \) could be improved by defining a semantic metrics \( m(\cdot,\cdot) \) on the concepts, so that a concept \( C_i \) not satisfying the not-null intersection condition can be added to the set \( \Pi(N) \) if exists another concept \( C_j \) such that \( m(C_i,C_j) < V \), i.e., the semantic distance between \( C_i \) and \( C_j \) is bounded by a threshold \( V \). As for the \( \Sigma(N) \) case, the context of e-learning systems could be a clarifying example about the application of the formal construct \( \Pi(N) \): a student, or a teacher too, can ensemble an e-course by selecting a set of topics, also coming from different scientific areas. So, when the user composes the e-course in a transparent manner, the underlying application logic should be able to define and execute the right knowledge extraction task (i.e., the right instantiation for the construct \( \Pi(N) \)).

Despite to the above general framework, for different application domains, such as people-finding systems (i.e., systems devoted to find people skilled in a certain role for very complex organizations), defining ad-hoc querying engines is mandatory.

From a technological point of view, a DKN network is an agent-based system. Each entity of the data layer is described in XML, also using meta-data for capturing data constraints, in order to allow cooperation and information sharing among heterogeneous components running on different software platforms.

4. A Reference Architecture for DKNs

In this Section we present our proposal about the implementation of DKN networks, as depicted in Figure 2.

As shown in Figure 2, the DKN Manager is the application server that is in charge to coordinate and manage all the knowledge nodes belonging to the DKN network. DKN Manager is the main component of the architecture and it contains the (Java) application logic that performs the server-side functionalities of the system.
The main software layers of the DKN Manager are the followings (see Figure 2):

- **KN Coordination Layer**: it implements a XML-based coordination/synchronization protocol among two or more nodes;
- **Knowledge Integration Layer**: it deals with the integration of the different knowledge sources on the basis of a model-oriented schema integration;
- **Services Differentiation Layer**: it aims to differentiate the exported services by their semantic domain (e.g., financial news are delivered to business men while last-minute travels information are delivered to young and student people);
- **Content Adaptation Layer**: it realizes the adaptation of the extracted knowledge taking into account the current user profile;
- **Device Adaptation Layer**: it is the software layer that performs the adaptation of the data coming from the Content Adaptation Layer on the basis of the current user device.

**User Behaviour Observer Agent** and **User Profiling Module** are the other main components of the proposed reference architecture.

**User Behaviour Observer Agent** is devoted to monitoring the user behaviour during the user/DKN interaction and collecting the parameters that will be used for clustering the user and assigning him/her to the appropriate user profile. The recent explosion of the technology and the consequent proliferating of smart, personal devices have made very hard this modelling task, since there are a lot of parameters that must be captured to ensure the quality of modelling.

User Profiling Module aims to build user profiles accordingly, starting from the outputs of the User Behaviour Observer Agent (see Figure 2). It holds a persistent data connection with the User Profiles DB, which is a relational database containing all the user profiles managed by the DKN network, and it updates the database each time a new user profile is detected.

We have used the very popular OLAP technology [8], coming from the Data Warehousing research community, for designing both the monitoring model and the user model. In our multi-dimensional model each dimension represents an observed parameter of the user model and the measures select the right XML data for the current user, according to his/her actual position in the multi-dimensional space. For instance, a possible entry of this model is:

\[ M\{\text{tourist, expert, PDA, 9 Mbps, MS Windows CE .NET, MS Internet Explorer CE .NET}\} = \{\text{\dymContent\schema.xsd, } \text{\dymPresentation\presentation.xsl}\} \]

This entry means that a user belonging to the cluster representing the tourist people and to the expert profile, having a PDA as connecting device with a bandwidth equal to 9 Mbps, having MS Windows CE .NET as OS platform and MS Internet Explorer CE .NET as web browser, accesses the knowledge exported by the DKN network by using the schema stored in the XMLSchema file schema.xsd, located in the directory `\dymContent\`, and the presentation rules stored in the XSL file presentation.xsl, located in the directory `\dymPresentation\`.

5. Personalizing Knowledge

In this Section we present our proposal about the knowledge personalization goal. As discussed in Section 4, knowledge personalization starts from all the information collected about the user during his/her interactions with the system.

Given a user \( U_i \) and \( M \) different subsequent sessions \( T = \{T_0, ..., T_M\} \), at session \( T_j \) the proposed approach chooses the right knowledge (i.e., data and services) on the basis of the current user profile \( P_j \), which corresponds to the profile of the user \( U_i \) evaluated according to his/her behaviour from the session \( T_0 \) until the session \( T_j \). We denote by \( W_{1}(T_{K-1}, T_K) \) the 1-step transaction from the session \( T_{K-1} \) to the session \( T_K \). More generally, we denote by \( W_{m}(T_{K-m}, T_K) \) the \( m \)-step transaction from the session \( T_{K-m} \) to the session \( T_K \).

Next, we provide the formal definition for our multi-dimensional user model. Let \( G = \{\rho_1, \rho_2, ..., \rho_{Q-1}\} \) be a set of \( Q \) observed parameters about user behaviour (e.g., \( \rho_q \),...
Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’04)

for our work. Respect to a

the User Profiles DB (see Figure 2).

such that \( E(\rho_q) \subseteq E(\rho_p) \ \forall \ q \in \{0, 1, ..., Q-1\} \).

Relation (1) represents our multi-dimensional user profile model.

Moreover, given a user \( U_i \) and a session \( T_K \), we define the behaviour of the user \( U_i \) in the session \( T_K \) as:

\[
B_{i,K} = \{b_0, b_1, ..., b_{Q-1}\}
\]

such that \( b_0 \in E(\rho_0), b_1 \in E(\rho_1), ..., b_{Q-1} \in E(\rho_{Q-1}) \).

Relation (2) represents our multi-dimensional user behaviour model.

Similarly, we define the behaviour of the user \( U_i \) across a \( 1 \)-step transaction \( W_i(T_{K-1}, T_K) \) as:

\[
B_{i,K-1\rightarrow K} = \{<b_{0,K-1}, b_{0,K}>, ..., <b_{Q-1,K-1}, b_{Q-1,K}>\}
\]

and the behaviour of the user \( U_i \) across an \( m \)-step transaction \( W_m(T_{K-m}, T_K) \) as:

\[
B_{i,K-m\rightarrow K} = \{<b_{0,K-m}, ..., b_{Q-1,K-m}, b_{0,K}>, ..., <b_{Q-1,K-m}, ..., b_{Q-1,K}>\}
\]

Profile change detection is the most important issue for our work. Respect to a \( 1 \)-step transaction \( W_i(T_{K-1}, T_K) \), we introduce the variation of the behaviour for a given user \( U_i \) during \( W_i(T_{K-1}, T_K) \) as:

\[
\Delta_{i,K-1\rightarrow K}(B_{i,K-1}; B_{i,K}) = B_{i,K} - B_{i,K-1}
\]

So, we can model a profile change for a user \( U_i \) belonging to the profile \( P_j \) during \( W_i(T_{K-1}, T_K) \) as follows. If \( \Delta_{i,K-1\rightarrow K}(B_{i,K-1}; B_{i,K}) = \emptyset \), then the user \( U_i \) still belongs to the profile \( P_j \). If \( \Delta_{i,K-1\rightarrow K}(B_{i,K-1}; B_{i,K}) = \emptyset \), then the user \( U_i \) still belongs to the profile \( P_j \); otherwise (i.e., \( \exists b_h \in P_j \land \exists P_z: b_h \not\in P_z \)), then the user \( U_i \) belongs to another profile \( P_z \). If \( P_j \) not exists, then a new profile is created and added to the User Profiles DB (see Figure 2).

Without loss of generality, we introduce the variation of the behaviour for a given user \( U_i \) during an \( m \)-step transaction \( W_m(T_{K-m}, T_K) \) as:

\[
\Delta_{i,K-m\rightarrow K}(B_{i,K-m}, ..., B_{i,k}) = \Delta_{i,K-1\rightarrow K} \circ \Delta_{i,K-2\rightarrow K} \circ \Delta_{i,K-3\rightarrow K} \circ ... \Delta_{i,K-m\rightarrow K+m-1}
\]

and the change profile detection on an \( m \)-step transaction is modeled by generalizing the task given for the change profile detection on a \( 1 \)-step transaction.

Next, we describe how the system performs the knowledge personalization task. According to the previous formal framework, in a given session \( T_{K-i} \), a user \( U_i \) belonging to a profile \( P \) holds a (previous) behaviour \( B_{i,K-i} \). So, when the user \( U_i \) accesses the system, session \( T_K \) starts and the User Behaviour Observer Agent begins to collect information about him/her (see Figure 2). Note that during the session \( T_K \), user \( U_i \) accesses knowledge by using values belonging \( B_{i,K-i} \) as entry for the OLAP-based multi-dimensional user model \( M \) (see Section 4). When \( T_K \) ends, the new behaviour \( B_{i,K} \) is defined and the system computes \( A_{i,K-i\rightarrow K}(B_{i,K-1}; B_{i,K}) \) in order to determinate if the user \( U_i \) still belongs to the same profile \( P \) or to another one. In any case, a \( current \) entry for the model \( M \) is defined and it is used to extract the \( current \) personalizing objects (i.e., the XMLSchema file and the XSL file – see Section 4), which will be used to personalize knowledge at the next session \( T_{K+1} \).

Knowledge personalization task for an \( m \)-step transaction is the generalization of the previous task provided for a \( 1 \)-step transaction.

6. Conclusions and Future Work

In this paper we have presented the definition of DKN networks and a proposal for their implementation. A DKN network is a network of devices (wired or wireless) that provide the right information to the right user at the right time and in the right format, taking into account the user device capabilities. At the current state of implementation, we have focused our work on the development of those system components devoted to personalizing the knowledge. A very promising approach is the adoption of OLAP technology in order to improve the capabilities and the granularities of the user models, both for the tracking/observing phase and for the profiling phase. To the best of our knowledge, there are not other similar approaches in this research area, and consequently, our proposal can be considered innovative.

Future work will be focused on two directions. On the one hand, we will work on the complete implementation of the proposed architecture and on testing and tuning it on real-life realizations, such as e-learning and e-procurement systems. On the other hand, we will concentrate on the exploitation of the fundamental distributed knowledge concept, by directing our research effort towards the integration of this concept with the recent proposals about pervasive and ubiquitous systems, adding to these ones a knowledge-oriented fashion.
7. References


