A Framework for Experience Management in Public Organisations

Simon C. Lambert, Alvaro E. Arenas, Alistair J. Miles*

Abstract

The Pellucid project developed an adaptable and customisable platform for enabling experience management in public organisations. A framework for experience management has been developed based on the generation of 'active hints' that are presented to the user according to working context. Working context encompasses both position in the work process and domain-specific characteristics. The paper presents this framework and describes the engineering process that was undertaken following the CommonKADS methodology.

Keywords*: Knowledge Management, Experience Management, E-Government, Workflow Management Systems, CommonKADS.*

1 Introduction

Organisational mobility is the pervasive movement or circulation of staff from one unit or department within an organisation to another. This is commonplace in public organisations, which may deliberately encourage it as a form of career development. It is clear that organisational mobility is not necessary a bad thing: inasmuch as mobile employees bring fresh ideas or experience of other areas, then the organisation can be enriched. Nevertheless, inevitably these employees will find it harder to perform as effectively as more experienced (static) staff, due to their relative lack of specific knowledge obtained from experience. Time must be spent in gaining familiarity, and although there might be training available, this is not sufficient in itself. It is these problems that the Pellucid project aims to address [13, 11].

Pellucid uses the metaphor of an intelligent assistant who looks over one's shoulder and answers questions one might have at a particular point of work [16]. The assistant detects that an employee is working in a particular process, offering knowledge resources that facilitate her/his work according to her/his expertise. To this end, the Pellucid platform integrates technologies such as autonomous co-operating agents, organisational memory, workflow and process modelling, and metadata for accessing document repositories. The platform was installed in three pilot sites: the management of publicly funded projects in the Mancomunidad de Municipios del Bajo Gualdaquivir (MMBG) in Spain; the installation and maintenance of traffic light plants in the Traffic and Mobility Management Department of the Comune di Genova in Italy; and the call center for management and resolution of fixed telephony breakdowns of the Consejeria de la Presidencia de la Junta de Andalucia in Spain. Such diversity of applications requires flexibility and adaptability as two important characteristics of Pellucid.

In this paper we describe the knowledge engineering and the knowledge level modelling undertaken in Pellucid, taking into consideration the adaptability of the system to different

* CCLRC Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.
{S.C.Lambert, A.E.Arenas, A.J.Miles}@rl.ac.uk

business processes in public organisations. We begin by introducing the three pilot applications in Section 2. Then, Section 3 shows how experience is managed in Pellucid and presents the *active hint* concept. The core of the paper is Section 4, which shows the whole design process following the CommonKADS methodology. Next, Section 5 describes how Semantic Web technologies were used in the project. Section 6 discusses some issues relevant to the customisation of the Pellucid platform. Section 7 relates Pellucid with other works. Finally, section 8 presents final remarks and future work.

2 Three Applications for Experience Management

Within the Pellucid project, three pilot applications have been chosen in order to test and to show the functionalities of the platform. The three end users involved in the project are:

? Mancomunidad de Municipios del Bajo Guadalquivir (MMBG), an association of local governments in the south of Spain, which selected as its case study the management of projects and services.

? Consejería de la Presidencia (CPRE-JA), a body of the regional government of Andalusia (Junta de Andalucía) in Spain, together with the company SADESI that operates its call centre for telephony problems. The management and resolution of fixed telephony breakdowns in the call centre was the selected case study.

? Municipality of Genoa, Mobility and Transport Directorate (CdG), Italy. The planning and installation of new traffic lights was selected.

These applications cover a wide range of processes existing within the public organisations in several sectors. In this way the Pellucid platform is tested by different varieties of end users and is addressed to several employees. Moreover they cover different ranges of organisational mobility. In the cases of MMBG and CdG, there are many different mobility scenarios, in that there are many ways in which staff can move from one position to another, whether vertically through promotion or horizontally from another unit. For SADESI, the mobility arises from the high level of staff turnover—a common situation in call centers generally.

All the processes have much experience to manage and also to capture and to return to new employees: sometimes the experience is technical, and other times it is more related to administrative issues and procedures. For example, it might relate to who to contact with enquiries about a particular case, or how long to expect a certain process to take before it needs checking, or particular factors to take into account in special circumstances.

In the case of MMBG, the management of public funded projects includes all the tasks to be performed from the very early stages of a project (definition of the idea, preparation of the proposal and submission to the funding authority) to the justification of the project's costs and activities, and the preservation of all the documents generated during the project execution. The main problems to be addressed are due to the lack of integration among the departments involved in the different project phases, and to the non-existence of a solid information structure to provide support for external and internal issues, such as contacting potential providers or preparing a proposal. This is however not simply an IT issue; it reflects the varied and ill-structured nature of the work done. Thus staff find themselves without support—not only new employees but also experienced staff who can be involved in very different proposals and projects.

Employees in the Call Center of SADESI have to adapt in a very short time to a very specific working environment, getting a small amount of 'on-the-job' training support. These employees suffer from a very high mobility degree, and it has been proven that, due to their working conditions, they typically stay in their position only 6 months as an average. This very unstable environment makes the need for systems for registering, keeping and exploiting efficiently the experience and knowledge of the employees before they leave the Call Center.

In the case of CdG, the application related to the installation of a new traffic light exemplifies the important role of experience in the effective execution of the task. This procedure involves many actors and is made up of several phases: a preliminary analysis following a request for a new traffic light; verification of information; design in detail the plant; definitive evaluation of all aspects of the project; definition of financial and administrative aspects; installation of the traffic light; and finally, testing and maintenance. A major source of knowledge management problems is the large number of actors involved, and the need for co-ordination among the several departments in order to get the optimal final result. Less experienced employees have less awareness of who they need to contact at what stage among the external actors, how this contact should be made (by email, by letter, whether there is a standard form for a letter, …), and how long the actor normally takes to reply. These are all examples of knowledge that an experienced employee will have acquired, and they should be shared effectively.

The main benefits expected from the Pellucid project are the improvement in efficiency and effectiveness due to the reduction of time spent, and leveraging of experience due to the movement of staff among different roles.

3 Experience Management in Pellucid

The experience management model in Pellucid exploits the experience sharing concepts expressed in [6]. It is based on two main ideas: every employee in an organisation is both a provider and a user of experience; and employees communicate their experience to a common medium, then retrieving experience in a future from this common medium. Pellucid aims to be such a common medium.

The Pellucid experience management model comprises three phases: *Capture and Store*, *Analysis and Presentation*, and *Experience Evolution*.

The *Capture and Store* phase is concerned with observing and storing experience in a particular context. There are three ways of capturing experience: analysing employees' actions and workflow events, analysing documents entered into the system, and by direct input from workers. Capturing experience from working actions and events is particularly beneficial in repetitive tasks; they are used to create common patterns that can be retrieved in the future in order to assist other employees. Documents constitute an important asset in an organisation. Metadata is added to documents, enabling the system to retrieve in an automatic way documents useful in a particular working context. The direct capture of experience from employees is carried out through free-text notes written by the employees themselves. This constitutes a good source of knowledge, particularly in the transmission of experience from experienced employees to novices.

The purpose of Pellucid is to support and enhance employees' performance by providing them with the knowledge required by the activity they are performing at the time they are actually performing the activity.

The *Analysis and Presentation* phase is concerned with providing such knowledge. To do so, the concept of an *active hint* is introduced, a representation of experience within the organisation. An active hint is triggered in a context and includes an action, a knowledge resource and a justification for the hint. The context is determined by the particular activity that is carried out by the employee at that time in a workflow system. An action corresponds to an atomic act on a knowledge resource, for example use a document template, read a document or a note, or consider a contact list. The justification gives to the employee a reason for the hint. The idea of active hints was borrowed from the DECOR project [1] and worked out in a somewhat different way in Pellucid. Let us now describe the stages included in this phase:

? An employee is performing a particular activity. The system tries to *match* the current working context with other context stored in the organisational memory. In general, every past context is stored with possible solutions (hints) to assist employees in that particular situation.

? Rarely (if not ever) a perfect match occurs. So the system should be able to *adapt* the previous context to the new one as well as the past solution.

? The system suggests the past solution (maybe adapted) in the form of *hint*.

? The user is free to follow or not the hint given by the system. The user should be able to communicate this decision to the system which:
  - o If the hint is followed, it also contains a judgment of the proposed solution (it worked well/badly)
  - o If the hint is not followed, it also contains a motivation for it (I did not follow it because ...)

? The system should store this information, classifying them maybe with keywords or other techniques.

The aim of *Experience Evolution* is updating the available experience. Due to the rapidly changing environment, experience may have only a limited lifetime. Invalid experience must be identified and removed or updated. To this end, the final Pellucid platform will include a set of methods and semi-automatic tools to allow knowledge engineers and expert users to update the experience stored in the organisational memory.

3.1 Active Hints

The main conveyor of experience in Pellucid is an active hint. Active hints present suggestions to the user to assist with the current activity. Figure 1 depicts a schematic example of an active hint in the context of a proposal evaluation when managing a project for the MMBG site application, while Figure 2 shows an example from the CdG application.

Pellucid offers a variety of active hints, which are returned by analysing the current context of an employee and using techniques such as document similarity. Thus, active hints management allows the experience dissemination within the organisation.

Context:	Proposal Evaluation
Action:	Consider
Resource:	List of People and Documents
Justification:	People in the list have evaluated similar proposals in the past, and they have used the associated documents for such activity.

Fig. 1. An example of an active hint for MMBG

Context:	New employee at Technical Staff level is designing planimetry Installation is located near school or hospital or railway station
Action:	Examine
Resource:	Topographic relief map for the zone
Justification:	Such locations have high pedestrian flows and need special consideration to accommodate it, particularly size of the sidewalk and availability of pedestrian crossing.

Fig. 2. An example of an active hint for CdG

4 Engineering Experience Management in Public Organisations

The analysis and design process in Pellucid followed the CommonKADS methodology [15] and it is fully explained in [4]. In CommonKADS, the development of a system entails constructing a set of models of problem solving behaviour in its concrete organisations and application contexts. The first phase in designing Pellucid has been the development of the organisational, task, knowledge and communication models for the case of a generic public organisation.

4.1 Organisational Model

The organisational model describes the organisation in a structured, systems-like fashion. It includes aspects such as identification of problems and opportunity areas, organisational structure and resources. All these components come into play and interact when a new knowledge solution is introduced into the organisation.

4.1.1 Identifying Knowledge-Oriented Problems and Opportunities.

In the case of public organisations, we have identified the following problems related to mobility of public employees:

- When a worker leaves the organisation or changes to another department, there is no mechanism for preserving her/his experience in the previous position.
- When a worker changes position within the organisation, due to organisational mobility, s/he does not usually receive training. The acquisition of the specific experience is often left to the initiative of the colleagues or the worker.
- New workers usually receive some sort of training when arriving in the organisation. The training is often planned to communicate to them the specific rules and procedures of the organisation. Many relevant aspects of the position are normally left out.

These problems present opportunities that should be considered in Pellucid design:

- Capture the experience about operational processes of public employees.

- Leverage the accumulated expertise of employees.
- Make tasks more efficient and reduce wasted time in searching for information.
- Other more efficient and uniform responses to the public from the organisation.

4.1.2 A General Business Process for Public Organisations

In general, the mission of public organisations is to serve the needs of the community [7]. Those needs are represented by a portfolio of services offered by the organisation. As business process, we have selected the processing of a generic service as illustrated in Figure 3.

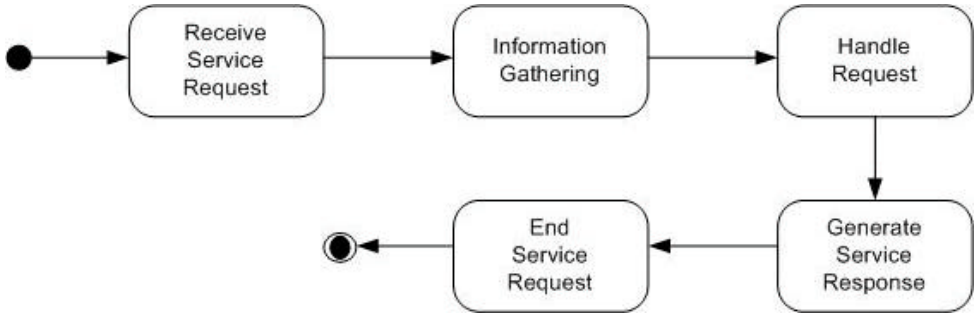

Fig. 3. Layout of a General Business Process

In Figure 3, Receive Service Request is concerned with the reception of service request. Once a service is requested, the next action is to determine which information is appropriate to answer the request. Then, the information needed to process the request is collected. Handling a service depends on the service itself, but here we analyse typical tasks such as writing reports, assessment of the service, etc. Finally, a response is generated for the customer and the service ends.

4.1.3 Describing Knowledge Resources

Pellucid assumes that the business process of an organisation is automated with a workflow management system. The system also creates and maintains an organisational memory which contains knowledge resources. At any point of work, an employee may request assistance of Pellucid, which offers then a hint indicating which actions could be executed and which knowledge resources are useful. The following are example of knowledge resources used by Pellucid:

? **Profile of Activity**. CommonKADS tasks correspond to the tasks involved in solving a service request. These tasks are divided into sub-tasks we have called activities. It is important to store information about each activity, such as its time criticality, knowledge grade, description, etc.

? **Profile of Employee**. The profile of an employee contains information relevant for the organisation, since from there it would be possible to determine her/his expertise and knowledge. Important components are the competences, capabilities and skills of each employee.

? **Documents**. Public organisations are document-centred, so this is an important knowledge resource. This resource denotes all documents owned by the organisation. It is

useful to indicate the different kinds of document available. We have identified the following kind of documents: manuals, working guidelines, work reports, assessments, meeting minutes, financial reports (spreadsheets), emails and images.

4.2 Task Model

In the context of CommonKADS, a task is a subpart of a business process that represents a goal-oriented activity adding value to the organisation, handles input and delivers desired outputs in a structured and controlled way, consumes resources, requires and provides knowledge and other competences, and is performed by responsible and accountable agents. The methodology includes textual templates to specify the tasks which includes information such as goal of the task, pre and post-conditions, objects handled, timing and control among others.

We have defined a catalogue of knowledge-intensive tasks that may be used in any of the tasks associated to the business process presented in Figure 1. This catalogue is based on the template of knowledge tasks proposed by CommonKADS in [15], which includes general tasks such as classification, diagnosis, assessment, monitoring, design and assignment among others. Pellucid catalogue includes the following tasks:

? **Classification of Documents.** Giving a document, this task allows one to know the classification of the document according to a topology. Such classification is useful in providing information/aid to users at any point in the business process.

? **Roadmap of Documents and Contacts Useful in a Process.** This task aims at determining which documents/contacts are useful in a particular process, when knowledge about that process is incomplete.

? **Monitoring Progress in Handling a Service.** Some processes are time-critical in the sense that the work should be done within a particular time. This task monitors the progress of such process, warning employees in case there is a risk of not ending the process in the specified time.

? **Resource Assignment.** Given the nature of a process, it could be necessary to assign some resources to it. This task suggests an employee an optimum assignment of resources to such process.

For instance, one of the most knowledge-intensive tasks in Figure 3 is *Information Gathering*. It requires determining the most appropriate information for the requested service and then obtaining such information. Determining the appropriate information may include generating a *Roadmap of Documents and Contacts* useful in that service. Figure 4 shows the task analysis worksheet for that task.

4.3 Agent Model

In CommonKADS, an agent can be human, an information system, or any other entity capable of carrying out a task. The agent model describes the characteristics of agents, in particular their competences, authority to act, and constraints in this respect.

There are three main elements in Pellucid: the employee of the organisation, a workflow system (WfMS) representing the business process of the organisation, and an organisational

memory containing the relevant information of the organisation. In general, Pellucid assists the employee at any point in the workflow system by providing useful information from the organisational memory. Agents in Pellucid reflect these components.

Task Generating Document/Contact RoadMap		
GOAL	This task aims to collect all document and contacts useful in a process. We can see this task as a particular case of Information Gathering	
FLOW	**Input tasks:**	Receive Service Request
	Output tasks:	Handle Request
OBJECTS HANDLED	**Input objects:**	Service Description
	Output objects:	List of documents and contacts useful to serve service
	Internal objects:	List of activities associated to the input service Context of each activity of the input service
CONTROL	**Precondition:**	Input service belongs to the service directory
	Postconditions:	Documents and contacts are represented by valid links to their position in the organisational memory
AGENTS	Information Search Agent	

Fig. 4. Task Analysis Worksheet for Generating Document/Contact Roadmap

The employee is represented by the Personal Assistant Agent (PAA), which is an interface agent serving and personalising information and suggestions to the user. There is a Personal Assistant Agent per employee in the organisation. The interaction of Pellucid with the workflow system is managed by the Monitoring Agent (MA). The whole functionality of the organisational memory (OM) is characterised by three agents: the Role Agent (RA), which acts as an interface between the Personal Assistant Agent and the organisational memory, the Information and Search Agent (ISA), in charge of searching and retrieving information from the organisational memory, and the Capitalisation Agent (CA), which generates new knowledge in the organisational memory.

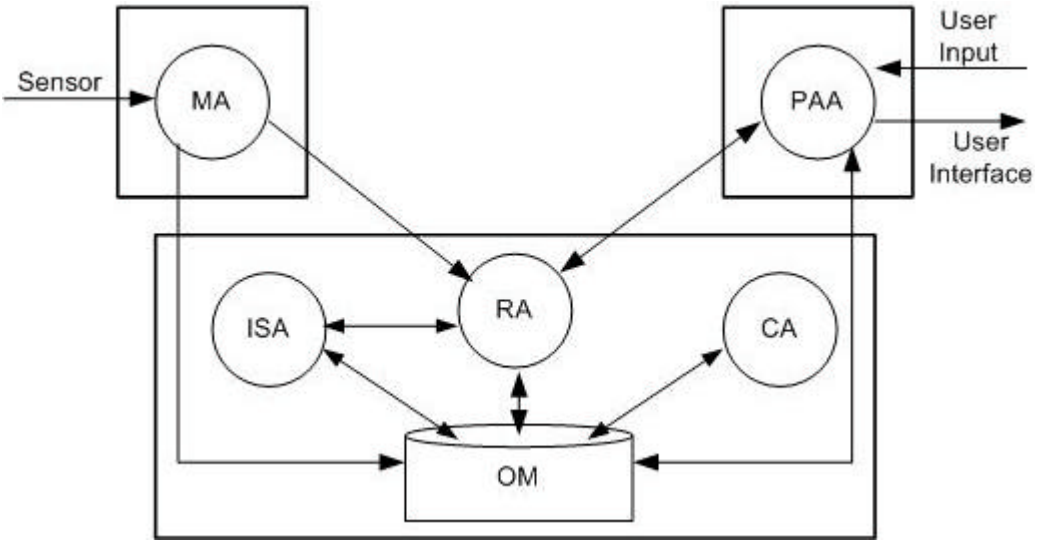

Fig. 5. General Agent Architecture of Pellucid

Figure 5 illustrates the general agent architecture of our system. The Monitoring Agent captures inputs from the workflow management system by means of its sensors and using an event-driven control regimen. The arrows in Figure 5 indicate information flow; hence the Monitoring Agent informs the Role Agent about received events and updates the Organisational Memory.

The Personal Asistant Agent manages the information provided to the human user by applying personalisation techniques; this agent includes functionalities such as presenting personalised information to the user, accepting comments/rating from the user about current task or selected information resource, and supporting administration of the user profile.

The core of the system comprises the Role Agent, the Information and Search Agent and the Capitalisation Agent. The Role Agent includes functionalities such as pushing newly discovered knowledge to Personal Assistant Agents of interested users, handling queries from Personal Assistant Agents, and delegating more complex or time-consuming queries to the Information Search Agent. The Capitalisation Agent creates global knowledge and reorganise historical data and knowledge data. It includes functionalities such as asserting new facts based on review of historical data from the organisational memory.

CommonKADS proposes templates for documenting the agent specifications. By way of illustration, Figure 6 shows the specification of Role Agent.

Role Agent Specification	
OBJECTIVE	Provide a user with a list of possible hints based on their current position in the business process
DESCRIPTION	The Role Agent generates active hints for the user based on her/ his current context. Some of the hints could require complex queries, which are delegated to the Information Search Agent
INVOLVED IN	Context matching; Generating active hints; Sending active hints
COMUNICATES WITH	Monitoring Agent, Personal Assistant Agent, Information Search Agent
KNOWLEDGE	This agent receives events from the Monitoring Agent indicating current activity of the user within the business process. It applies heuristics to determine similar contexts, based on activities profiles stored in the organisational memory, and to generate proper hints to the user

Fig. 6. Specification of Role Agent

4.4 Knowledge Model

The knowledge model specifies the knowledge and reasoning requirements of the prospective system. It includes the domain knowledge model, which specifies the knowledge and information types we want to talk about in the system, and the task knowledge, which describes the goals an application pursues and how these goals can be realised through a decomposition into subtasks and inferences.

4.4.1 Domain Knowledge Model

We present here a fragment of the domain knowledge model and refer the reader to [14] for a more complete model. Most constructs of the domain knowledge model are similar to the ones used in modern object-oriented data models. Following [3], we use a notation as close as

possible to UML to represent the domain-knowledge constructs and express them later in the Web Ontological Language OWL.

In general, public *Organisations* have as mission to serve the needs of a community through a set of *Services* which consists of *Activities* and involves *Employees* [7]. Each activity requires *Skills* from the employee in order to be performed efficiently. An employee is chosen to work within a particular service on the basis of a *Role* played within the organisation. However, the role of an employee may change dynamically according to current needs. The employee who is subject to frequent changes of role within the organisation is a mobile employee (see Figure 7).

.

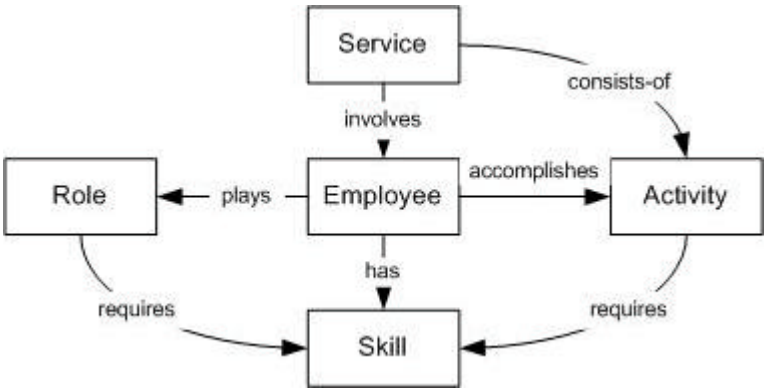

Fig. 7. Fragment of Domain Knowledge

Generally, each business process can be divided into smaller steps called *Activities*. To accomplish an activity, an employee undertakes some *Actions*. It is worth noting that actions are not part of the business process, since different people can accomplish an activity in different ways. An activity depends on the problem to be solved and such information is captured in the concept of *Context*. The purpose of Pellucid is to support and enhance employees' performance by providing them with the knowledge required by the activity they are performing at the time they are actually performing the activity. To do so, the concept of *Active Hint* is introduced, explained previously. Experience can be seen as knowing what to do in particular circumstances. The circumstances corresponding to the context and knowing what to do is characterised by the action and resources needed in that action (see Figure 8)

.

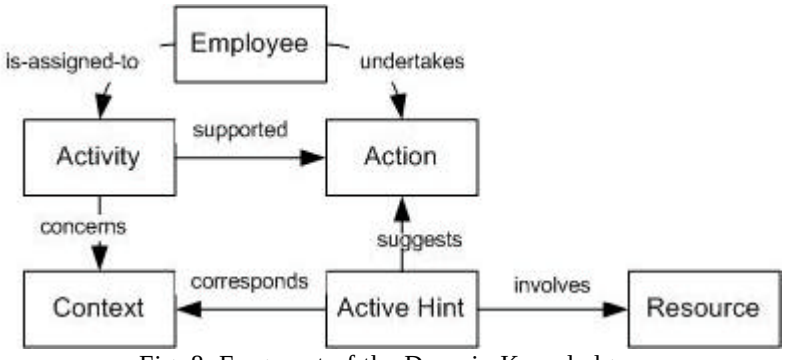

Fig. 8. Fragment of the Domain Knowledge

All elements modelled in the domain knowledge model constitute a general ontology for experience management. This ontology is used as a mechanism to express the main concepts of the system and their relation as well as to infer knowledge.

4.4.2 Context Modelling

Context is generally regarded as a major concept for the meaning of knowledge. As shown in the previous subsection, it is necessary to represent and handle the working context in order to manage the active hints for fulfilling the purpose of Pellucid.

The approach used is that of context-based retrieval information [5]. The working context is described by the position in the business process (starting or completing a task, opening a document, …) and domain-specific characteristics. In each of the Pellucid pilot application, a workflow is defined and implemented. This may be quite flexible—it does not have to be a rigid procedure. The domain is directly related to the domain application, i.e., call centre, traffic light installation and project management. Therefore, the instance in the workflow ontology is an incidence resolution, a dossier and a project. Figure 9 illustrates part of the working context modelling in the framework of each pilot application. Note that a wide variety of working domains may be represented within the Pellucid ontology of working context. At the higher levels the ontology is domain-independent; all work domains with a reasonable degree of structure will have actors and activities. At lower levels domain-specific concepts appear. The filling out of the ontology is part of the task of customisation.

Working Context modeling				
Concepts	**Representation**	**Examples**		
		Call centre	**Traffic light control**	**Project management**
Business Entity: Actor	Employee which starts an wfActivity in order to manage the wfInstance	*Id_agent*	*Id_officer*	*Id_project_manager*
WfActivity	The current stage of the process	*Intervention_ request*	*Preliminary_analysis*	*Proposal_submission*
WfInstance	The studied concept from the application	*Id_incidence*	*Id_dossier*	*Id_project*
Concept1	Other domain or workflow concepts used as relevant descriptive information in order to compute the similarity measure.	*Id_site*	*Id_crossing_road*	*Id_topic*
Concept2		*Id_severity*	*Id_number_lanes*	*Id_type_document*
Concept3		*Id_type_site*	*Id_existing_traffic_ light*	*Id_role_actor*
Concept4, etc		*Id_operator*	*Id_applicant*	*Id_objective*

Fig. 9. Modelling Working context for the Pellucid Pilot Applications

4.4.3 Task Knowledge Model

The task knowledge model defines the strategies that will be used to achieve the main goal of a task. Task knowledge is typically described in a hierarchical fashion: top level tasks such as *Generate Contact/Document Roadmap* are decomposed into smaller tasks, which in turn can be split up into even smaller tasks. At the lowest level of task decomposition, the tasks are linked to inferences - a primitive reasoning step - and input/output functions.

Let us examine tasks *Generate Contact/Document Roadmap*. The general specification presented in Figure 4 gives an informal description of the goal of the task and the relation

between input and output. It is worth mentioning there are no domain-dependent terms in such specification. This task can be seen as a particular case of an assignment task (cf. [15], chapter 6) in which we are linking (assigning) an activity to people and document resources taking into consideration some constraints such as the grade of knowledge of people - their expertise - and the relevance of the documents used. The method used to realise this task is summarised in Figure 10. In this method, decompose, determine and assign correspond to inferences. They are implemented using the ontology presented previously in the domain knowledge model. obtain and present correspond to input/output functions.

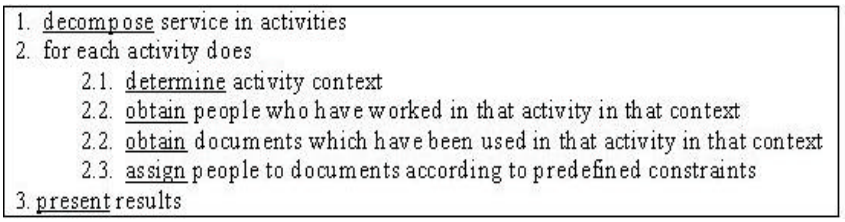

Fig. 10. Task Method of Generating Contact/Document Roadmap

5 Semantic Web Technologies in Pellucid

This section describes some of the finer detail of the design implementation of the Pellucid platform, and focuses on how Semantic Web technologies were used to achieve the desired experience management functionality.

The Resource Description Framework (RDF) and the Web Ontology Language (OWL) are cornerstone semantic web technologies, and are both W3C Recommendations. RDF provides a basic assertional language that is intended for making assertions both about a distributed network of information resources, and about abstract and physical concepts such as people, organisations, workflows etc. Moreover, RDF provides some basic and well-supported solutions for the handling of distributed assertion sets ('graphs'), allowing decentralised metadata repositories and knowledge bases to be integrated across an organisation. OWL provides an additional logical framework, which when layered on top of RDF becomes a well-featured, distributed knowledge representation language; distributed both in the sense that a network of resources can be described, and in the sense that the assertions or 'facts' themselves can be distributed and sensibly merged. OWL supports a useful set of logical inferences, which enables the logical dependencies between different parts of the knowledge model to be captured and effectively managed.

RDF and OWL were used as the basis for all agent communication in the Pellucid platform, and as the basis for the organisational memory. They were chosen for these roles because the essence of agent communication is the ability to communicate facts or assertions, and to ask questions (i.e. query another agent's knowledge base). RDF/OWL provides a language for expressing facts, and RDF/OWL toolkits support the serialisation and parsing of RDF/OWL serialisation languages such as RDF/XML. RDF/OWL toolkits also implement a query interface with some sort of RDF query language, which at the time of Pellucid depended upon the toolkit chosen, but which is now close to standardisation as 'SPARQL'. Thus much of the agent functionality could be implemented directly via standard RDF/OWL toolkits. Similar functionality is required of the OM, because facts or assertions need to be loaded, removed, and queried. Again standard RDF/OWL toolkits provide off-the-shelf solutions. Furthermore, the ability to integrate and merge assertions from multiple

sources, supported by RDF/OWL, is an essential part of managing the distributed information environment of any organisation.

More detailed example of the use of Semantic Web technologies in Pellucid are shown in [12].

5.1 The Organisational Memory

The organisational memory plays a central role in Pellucid. It was conceptually divided into two parts: one containing the current state of the organisation, and another one containing the history of all organisational events.

The description of the current state of the organisation is implemented as an RDF model. This model is called the *current state model*. An RDF model is a set of RDF statements. An RDF statement is an assertion of the form *(subject, predicate, object)*. Each RDF statement in the current state model represents an atomic assertion about the state of the organisation. These assertions combine to constitute a set of facts about the organisation at the present moment in time. Typically, the current state model will include a representation of the organisation's structure, its employees and their roles, a description of any working processes including their state, and descriptions (metadata) about knowledge assets (documents and contacts) in the possession of the organisation. The Pellucid generic ontology encoded in OWL provides the basic *vocabulary* for creating an RDF description of the organisation. Each domain specific ontology typically extends the Pellucid generic ontology to include vocabulary for describing domain specific aspects of the organisation.

The history of the organisation is also implemented as an RDF model. This model is called the *events model*. The events model stores a description of all the events that have occurred in relation to the business of the organisation, including for each event who or what was involved, and what the consequences where (i.e. what, if any, aspects of the organisation changed).

It is possible to represent an organisational event as three sets of RDF statements. The first set of statements is a description of the event itself, including the date and time at which it occurred, and who and/or what was involved in its cause and/or consequence. The second set of statements is the set of atomic assertions about the organisation that became true as a consequence of the event (*positive consequences*). The third set of statements is the set of atomic assertions about the organisation that were no longer true as a consequence of the event (*negative consequences*).

As each event is stored in the events model, RDF reification of statements is used to make assertions about which statements are the positive and negative consequences of the event being added.

To support historical analysis of events, it is necessary to be able to recall information about the state of the organisation immediately before each event occurred. However, storing a complete copy of the current state model with each event would be impractical. Therefore, each event is stored with a small set of *context features*, which are those features of the organisations state immediately before the event occurred that are deemed to be relevant to the occurrence of that event. The context features, as with the positive and negative consequences, are implemented as a set of reified RDF statements.

The Jena 2.0 Semantic Web Framework [8] was used to generate and manipulate the RDF models of the OM.

5.2 Organisational Dynamic

5.2.1 Generating Events

In the Pellucid agent architecture, the Monitoring Agents (MA) collect information about the state of the organisation by monitoring the organisation's applications, process and databases. The information received by an MA from its sensors is *interpreted*, i.e. is converted into an RDF description of an organisational event. This RDF description is then serialised, and passed as a message to a Role Agent (RA).

An example of a serialisation of an RDF description of a simple organisational event is presented in Figure 11. This is an event in which an employee opens a document using Microsoft Word 2000. The single (positive) consequence of this event is that the employee ('Pete') has an open file on his desktop. Events relating to use of documents can be a valuable resource, because knowing what documents are useful in what context is an important element of experience.

The RA receiving this message parses it into an RDF model. Role Agents have responsibility for event handling, which is the process by which the OM is updated.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
        <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
        <!ENTITY pel "http://www.clrc.ac.uk/pellucid/ontology/owl/generic/1.0">
        <!ENTITY clrc "urn:clrc/">
]>
<rdf:RDF xmlns="&pel;#" xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;" xml:base="&pel;">
        <EventFileOpened>
                <occurred>15-Dec-2003 12:50:46</occurred>
                <involvedActor>
                        <User rdf:about="&clrc;user/Pete"/>
                </involvedActor>
                <involvedFile>
                        <ComputerFile rdf:about="file:///folder/guide-v03.doc"/>
                </involvedFile>
                <involvedApplication>
                        <Application rdf:about="#MSWord2000"/>
                </involvedApplication>
                <add>
                        <rdf:Statement>
                                <rdf:subject rdf:resource="&clrc;user/Pete"/>
                                <rdf:predicate
                                        rdf:resource="#userOpenFile"/>
                                <rdf:objectrdf:resource=
                                        "file:///folder/afile.doc"/>
                        </rdf:Statement>
                </add>
        </EventFileOpened>
</rdf:RDF>
```

Fig. 11. RDF Serialisation of a Simple Organisational Event

5.2.2 Event Handling

When a Role Agent receives an event, it invokes a fixed set of *event handlers*. The first event handler is the *state manager*, the second is the *context selector*, and the third is the *event stack*. After being processed by these handlers, the event is then passed on to any further *event listeners* that have been registered with the OM.

The *state manager* uses the positive and negative consequences of an event to update the current state model. Quite simply, the statements representing the positive consequences are added to the current state model, and the statements representing the negative consequences are removed from it.

As mentioned above, each event is stored along with a set of *context features*. Because the set of relevant context features will depend heavily on the *type* of the event, each event type is defined with its own *context selector* properties. Each of an event type's context selector properties defines an RDF query that, when applied to the current state model, will extract the relevant context features for that type of event.

The *event stack* receives and stores all events in the events model. It orders all events according to the date and time at which they occurred, to support historical analysis functions.

An *event listener* is any object interested in being notified on the reception of an event by the OM. Any number of event listeners may be registered with an OM.

5.2.3 Firing Active Hints

Role Agents also build and maintain *hint engines*, which trigger the creation and delivery of active hints to users when the user enters an appropriate working context. An OM may have any number of hint engines bound to it at any time. This allows for different underlying mechanisms of hint triggering to be used concurrently.

The *rule-based hint engine* uses forward rules to trigger hints when a particular set of statements occurs in the current state model. The forward engine used is the Jena2 ReteReasoner.

The *event-based hint engine* uses the reception of certain event types to trigger hints. It stores a set of mappings from event patterns (implemented as RDQL queries) to hint types. Thus, when an event matching a particular pattern is received, the corresponding hint is triggered.

After a hint is triggered, it is built and then sent to the Personal Assistant Agent of the appropriate user.

6 Pellucid in Practice

6.1 An Example of Pellucid in Action

In order to make the ideas presented in the previous sections more concrete, a concise example is shown in Figure 12 of the process of hint generation from working context. It is given as a sequence of transactions between the agents of the Pellucid architecture, expressed informally in order to make clear the basic principles.

1. Employee starts task of proposal evaluation
   Obviously the system must be able to detect some employee behaviour. This is achieved through the workflow system.
2. Workflow system reports event to MA
3. MA translates event into terms of ontology and passes to RA
4. RA writes event to OM
5. Event is stored with features extracted from current context
   Features include expertise of employee, specific information about proposal worked on.
6. Event listener for hint engine reacts to event and triggers hint builder
   The generic hint type triggers when a new activity is started
7. Hint builder constructs hint to advise employee that previous documents were consulted at the same stage in similar proposals
8. RA requests the ISA to search the OM and/or external repositories for documents with specified properties
   For example, certain keywords or ontological annotations, or used by other employees at the same stage in other process instances.
9. ISA returns set of documents
10. The completed hint is passed to the PAA
11. PAA presents the hint to the employee
    Taking into account their preferences (simple user modelling)

Fig. 12. Sequence of hint generation in Pellucid

This example is of just one kind of active hint, albeit a very generic hint which is applicable across a range of applications. One of the strengths of the Pellucid design is the flexibility inherent in the hints. Very generic hints such as the one just shown may be applied in very different applications; the only requirement is that there are defined steps in the work process and that these steps have a history of other employees having performed them using certain documents that are still accessible. Of course, this does not mean that such hints will necessarily be valuable in all applications—the job of customising the Pellucid platform includes requirements acquisition to determine where the benefits from experience management will be felt.

Free-text notes, which at first sight appear to be a very different kind of hint, as their content is entered by users themselves, in fact fit within the same framework. The text of the note itself is regarded as a resource for retrieval and presentation to the employee. Thus the same mechanism is available for different applications. In addition, specially crafted types of hint may be constructed for particular applications. This can be seen as a knowledge engineering job, rather like building an expert system for experience management.

6.2 Obtaining Context from External Systems

It is desirable that the working context should be acquired automatically. The aim is to make the system as unobtrusive as possible (at least in the capture phase), so it is necessary to trade this off against completeness. This means that, even though it might have been possible to ask the users to input data directly into the system, the choice was made to have less data, but collected automatically.

The reasons for this choice are:

- The system is aimed also at less experienced employees (in fact this is one of the main reasons to have this system installed) and it is not acceptable to ask information of such an employee when he/she does not already know the context him/herself.

- It is difficult, or impossible, to justify the extra work imposed by the system on experienced employees when the help which the system is building is not primarily directed to them.

In other words less experienced employees are the principal targets of the system, but they are not the principal sources of expertise. In order to collect information from experienced employees it is necessary to resort to automatic collection. An exception is the case of free-text notes which are entered directly by employees, but here the context of the note is acquired by the system itself.

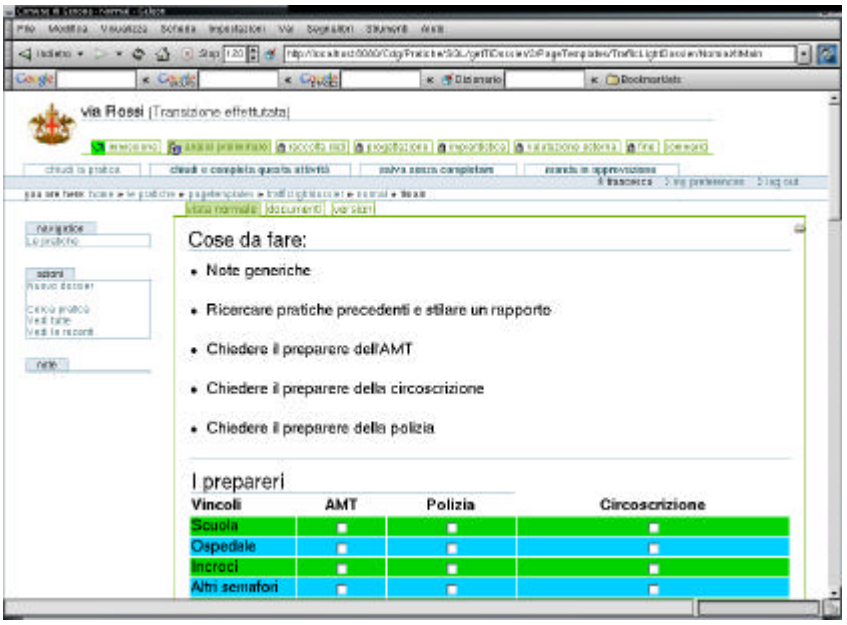

Fig. 13. Screenshot of the workflow tracking system for the CdG Application

To collect information about the actions and type of work of the employees, the use of some kind of workflow tracking system is the best trade-off. This is an auxiliary system whose purpose is to collect context information for the Pellucid system, though it may have other benefits for the user. A conventional workflow management system may be used for the purpose, though a simpler, more flexible system can be developed. To integrate with tracking system or WfMS the WAPI interface has been adopted, which is a standard. In this way the only work has been to build a bridge that converts the WAPI calls of the WfMS to XML-RPC calls, which is the standard way to interact with Pellucid. Figure 13 shows a screenshot of the user interface of the WFMS developed for the CdG application.

6.3 User Interface Issues

As said before, the capture phase is not observable by the user. Pellucid seems to be an 'output-only' machine. For the output too the 'least obtrusive' approach has been chosen. In fact Pellucid output is customizable along with everything else in the platform. The Pellucid core, in fact, does not care about the user interface but returns results in a 'format-free' way. The front end is then free to be built with a look and feel to suit the end user.

For example if the application is Web based (as in the case of the Comune di Genova), it is possible to build a HTML transformation so as to have a web window which integrates with the

application. An alternative is for a separate window dedicated to the Pellucid interface, as shown in Figure 14.

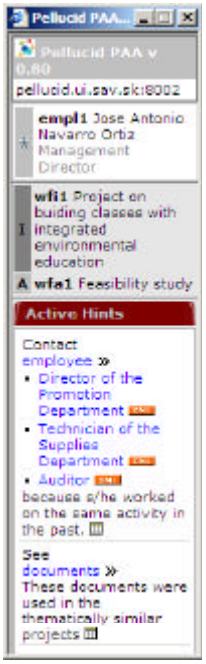

Fig. 14. An Example of a Free-Standing User Interface for the Pellucid System

6.4 Customisation In Pellucid

Customisation of the Pellucid platform consists in adapting and reusing elements, the type of reuse depending on the nature of the element itself: for example, the software representing an agent class might be stripped down into a template for future application, and/or provided as an example to be modified. The general reusable elements include the following:

- ontologies (general and domain-specific);
- general user interface (Web application);
- external system interfaces (to document repositories and workflow systems);
- the agents themselves;
- the organisational memory.

Figure 15 indicates a preliminary view of how the elements can be provided in the platform, and the corresponding kinds of customisation that will be applicable.

Element provided in platform	**How customized**
Core (e.g. of ontology)	Expand/Populate/Possibly adapt
Template (e.g. of agent class)	Fill in/Populate
Prior case (e.g. of agent class)	Adapt
Building blocks (e.g. of org. memory)	Assemble
Guidelines (e.g. for interfacing to document repositories)	Implement
Specifications (e.g. for interfacing to workflow management system)	Implement

Fig 15. Customisation of the Pellucid Platform

The Pellucid system has been successfully integrated with the three pilot applications of the project, and they were populated with a range of active hints and provided with user interfaces and interfaces to workflow systems. The emphases of each pilot application are somewhat different: at the Comune di Genova, free-text notes; at MMBG, hints about previously used documents and contacts; at SADESI, hints tailored to the specific needs of the call centre environment. Nonetheless they all fit comfortably within the general framework.

7 Related Work

Pellucid has developed an approach highly influenced by the CommonKADS methodology [15], adapting elements of other approaches such as management of organisational memories [10], experience management [6] and active hints [1].

Pellucid can be seen as an example of an Electronic Performance Support Systems (EPPS) [9], systems that aim to support and enhance users' performance by providing them with the knowledge required by the task they are performing at the time they are actually performing the task. Other examples of EPPS are the EULE system [Reimer00], the VirtualOffice and KnowMore systems [2] and integration of knowledge and business processes [16].

The EULE system aims at offering assistance in the office work in the domain of insurance business [Reimer00]. It includes a business-process modelling framework which starts with a high-level structure of business processes that span various organizational units, and then gets broken down into more and more local views which at the same time become more detailed, until at the most detailed level EULE office task representations are obtained. The system is coupled with a workflow system by linking a EULE office tasks to working steps of a workflow. When reaching a working step that is associated with a EULE task, the user can request EULE assistance and obtain then the information missing at that specific point.

The VirtualOffice and KnowMore projects aim to support knowledge-intensive activities by providing automatic access to relevant information [2]. Each activity belongs to some comprehensive business process which is explicitly modelled and enacted by some WfMS. The activities are to be supported based on an available information space, which contains information sources of various types and characteristics together with suitable access structures. A central component is an intelligent assistant, which bridges between the information space and the knowledge-intensive activities and performs a process identification job (similar to the context identification job carried out by Pellucid) in which the system detects the particular circumstances of a process. The Virtual Office tool integrates paper-based information into workflow using a document management system for information extraction, following a request from the workflow. The KnowMore project focuses on delivering goal-specific information in a proactive way by analysing the incoming stream of information that the organisation receives. These works were further extended in the DECOR project [1].

The work of Staab and Schnurr in [16] is close to our work in putting an intelligent assistant to work within a business process environment. It also exploits the inferential power of ontology-based retrieval on top of the Ontobroker software, using a notion of context-based views for coupling workflow and retrieval. In building the information system, they start with an analysis process that explores the interdependence among the documents employed in the business process. Then, domain-specific information is added by including domain ontologies describing the content of documents, and contextual information. During the execution phase, the system acts as

facilitator for sharing, creating and retrieving knowledge, providing users for active help as a response to their request.

8 Conclusion

The Pellucid project has shown that it is possible to develop a generic, customisable platform for creating experience management applications. The idea of active hints has proved highly flexible and capable of encompassing a wide variety of forms for presenting experience to public employees. The hints may be instances of a general type, specialised to the current working environment of the employee, or free-text notes entered by employees themselves and retrieved for others when the working context is similar.

Two of the challenges of the Pellucid project have been the representation of working context and the accumulation and effective use of knowledge in an organisational memory. Each of these presents trade-offs. For working context, the trade-off is between precision (implying that the Pellucid system has a full knowledge of the user's working situation) and the desire not to burden the user with extra tasks (such as manually specifying their current context). For the organisational memory, there is a similar trade-off between size and growth of the memory, and the effectiveness of the reasoning performed on its contents. The potential for 'forgetting' memory contents is important, since experience must be kept up to date.

Acknowledgements

This work was supported by the IST Programme of the European Commission, Project number IST-2001-34519. The authors are grateful to other project partners for helpful comments. Pellucid consortium comprises ACC CYFRONET-AGH in Poland, Institute of Informatics of Slovak Academy of Sciences in Slovakia, Sadiel S.A. in Spain, Softeco Sismat SpA in Italy, Consejeria de la Presidencia de la Junta de Andalucia in Spain, Comune di Genova in Italy, Mancomunidad de Municipios del Bajo Gualdaquivir in Spain, and SADESI in Spain.

References

[1] A. Abecker, S. Dioudis, L. van Elst, C. Houy, M. Legal, G. Mentzas, G. Papavassiliou. Enabling Workflow-Embedded Organizational Memory Access with the DECOR Toolkit'. In Dieng-Kuntz and Matta [10].

[2] A. Abecker, A. Bernardi, H. Maus, M. Sintek, C. Wenzel. Information Supply for Business Processes: Coupling Workflow with Document Analysis and Information Retrieval. *Knowledge-Based Systems* 13(5) (2000) 271–284.

[3] A.E. Arenas, B.M. Matthews. An Experiment on Modelling Agents and Knowledge for the Semantic Web. Eleventh International Conference on Artificial Intelligence: Methodologies, Systems and Applications. Lecture Notes in Artificial Intelligence, Vol. 3192. Springer, 2004.

[4] A.E. Arenas, S.C. Lambert, A.J. Miles. Engineering Knowledge-Intensive Tasks in Public Organisations. EU-LAT Workshop on E-Government and E-Democracy, Chile, 2004.

[5] T. Bauer, D. Leake. Exploiting Information Access Pattern for Contex-Based Retrieval. Proceedings of the 2002 International Conference on Intelligent User Interfaces, ACM Press, 2002.

[6] R. Bergmann. Experience Management, Lecture Notes in Artificial Intelligence Series, Vol 2432. Springer, 2002.
[7] J. M. Bryson. Strategic Planning for Public and Nonprofit Organizations. Jossey-Bass Public Administration Series, 1995.
[8] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson. Jena: Implementing the Semantic Web Recommendations. Technical Report HPL-2003-146, Hewlett Packard Laboratories, 2003.
[9] K. Cole, O. Fisher, P. Saltzman. Just-in-Time Knowledge Delivery. *Communications of the ACM* (1997) 40(7).
[10] R. Dieng-Kuntz and N. Matta, editors. Knowledge Management and Organizational Memories. Kluwer Academic Publishers, 2002.
[11] S.C Lambert, A.E. Arenas, S. Delaitre, J. Mena Raposo, P. Ferrentino, M. Majewska, K. Krawczyk, M. Fassone, V. Procopio.A Framework for Experience Management in e-Government: The Pellucid Project. Electronic Journal of e-Government 2 (3) p.167-176 F. Bannister (Eds), Academic Conferences Limited, 2004.
[12] S.C. Lambert, A.E. Arenas, A.J. Miles. A Semantic Web Approach to Experience Management in Public Organisations. EKAW 2004 Workshop on Knowledge Management and Semantic Web. England, 2004.
[13] S.C Lambert, S. Stringa, G. Vianno, J. Kitowski, R. S lota, K. Krawczyk, M. Dziewierz, S. Delaitre, M. B. Oroz, A. C. Gomez, L. Hluchy, Z. Balogh, M. Laclavik, M. Fassone, and V. Contursi. Knowledge Management for Organisationally Mobile Public Employees. In Knowledge Management in Electronic Government: 4th IFIP International Working Conference, KMGov 2003, volume 2645 of Lecture Notes in Computer Science, pages 203–212. Springer, 2003.
[14] J. Kitowski, K. Krawczyk, M. Majewska, M. Dziewierz, R. Slota, S.C. Lambert, A.J. Miles, A.E. Arenas, L. Hluchy, Z. Balogh, M. Laclavik, S. Delaitre, G. Vianno, S. Stringa, P. Ferrentino. Model of Experience for Public Organisations with Staff Mobility. 5th Working Conference on Knowledge Management in Electronic Government - KMGov2004. Lecture Notes in Artificial Intelligence, Vol. 3035, Springer, 2004.
[15] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Vand de Velde, and B. Wielinga. Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press, 2000.
[16] S. Staab and H. P. Schnurr. Knowledge and Business Processes: Approaching an Integration. In Dieng-Kuntz and Matta [10], pages 75–88.