Ontology-Based Experience Management for Public Organizations

Giang Nguyen T., Michal Laclavik, Zoltan Balogh, Emil Gatial,
Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
giang.ui@savba.sk

Alvaro Arenas
CCLRC Rutherford Appleton Laboratory
Chilton, DIDCOT, OX11 0QX, UK

Abstract

In this paper, the view into the ontology-based experience management in the Pellucid project is presented. The result of the project – the Pellucid platform works in cooperation with a underlying system, that reports workflow events for employees in public organizations. Experiences produced by employees are captured, stored and re-used as useful help for novice or other employees in public organizations. This can relieve and improve work effectively and observe important knowledge when a experienced employee leaves the organization. The paper also presents the platform architecture, its functional design and the brief platform evaluation.

1. Introduction

Pellucid - the Platform for Organizationally Mobile Public Employees is the 5FP European Project RTD IST-2001-34519 - is a project [1] concerned with knowledge management for public employees, specifically for those who are organizationally mobile, moving from one department or post to another. To assist these employees, it focuses on a subtype of knowledge management: experience management. The project is developing a customizable software platform for developing experience management systems in a wide variety of organizations. It integrates several leading-edge information technologies, including agent-based system development; organizational memory; ontology for representing information and knowledge; information retrieval and automatic information extraction; and integration with secondary applications such as Workflow Management/Tracking Systems (WfMS/WfTS).

The overall aim of the project is to develop an adaptable platform for assisting organizationally mobile employees, in effect to help their works in a public

Witold Abramowicz (ed.), Business Information Systems, Proceedings of BIS 2005, Poznań, Poland

organization. This will improve organization effectiveness and efficiency by formalization, recording, storage and preservation of experiences and support workers during integration in a new role by giving access to specific experience accumulated in the past. In public organizations, employees are various: experienced or inexperienced, novice (new hired) or moved from other positions. Experienced users have knowledge and experience in given workflow activities and in given situations. Experience shows an approach of executing activities that may be the best among the existing used ones …

2. Experience management in the Pellucid platform

2.1. Ontology base for experience management

Important background of the Pellucid design is its ontology, which defines structure and relationships among experience entities. The ontology is the main mechanism used for the representation of information and knowledge, definition of the meaning of the terms used in the content language and the relation among these terms. Ontology is essential to the whole work of the Pellucid platform. Agents communicate using ontology structure, store and retrieve data organized and described by ontology to/from OM.

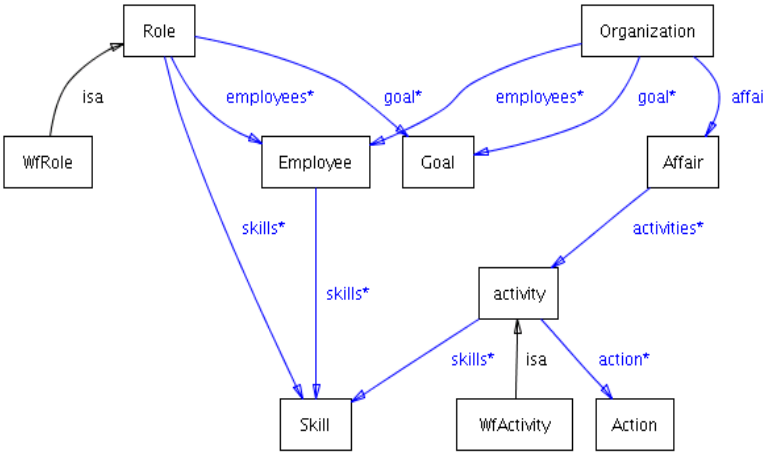

Figure 1. Generic ontology – Organization Model

The analysis and design of the Pellucid ontology includes a contextual and conceptual modeling, where aspects of a generic public organization are modeled in fluency with the CommonKADS methodology [2]. The Web Ontology Language [3] describes hierarchical relationships, property relationships, equivalent/disjoint concepts, cardinality constraints, etc.

When the Pellucid system is designed for three pilot site applications with extension possibility for next similar applications, the Pellucid ontology is composed from the generic ontology (unitary framework which contains generic features and general slots for building domain ontology for all three pilot applications and/or other applications) and domain ontology (defines terms for application, its concepts and dependencies and slots for concrete properties). The life-cycle of Pellucid ontology are:

- "kick-off": ontology requirements are captured and specified, competency questions are identified, potentially rebuilding ontologies are studied and a first draft version of the ontology is built;
- "refinement": a mature and application-oriented ontology is produced;
- "evaluation": the requirements and competency questions are checked, and the ontology is tested in the application environment;
- "ontology maintenance": ontology updating a extending by experts

Pellucid ontologies are implemented to support specific aspects of an organization (traffic-light installation, breakdown in telephone service, project management). The generic ontology defines basic, general and common features for all pilot site applications. Its classes are: action, active hints (composed of context, resource, action and explanation), domain, event (consists of employee, action, resource, and context), joint AH (enables to compose more complex hints), resource, timestamp, workflow activity and their properties.

Domain ontology is built individually for each (e.g. pilot) application. Each pilot site application contains following additional classes:

- CdD (traffic light establishment - Italian pilot site application): Applicant, Data, District, Intersection, Lane, Movement, Phase, Preparere, Simulation, Street, Traffic_Light_Part, Traffic_Light_Plant, Traf-fic_light_Timing;
- MMBG (telephone incident report application - Spanish pilot site application): Generic Term, Organization, Comity, Council, Project, Role, Term, Theme, Type of Document;
- SADESI (review and evaluation of projects - Spanish pilot site application): Generic Term, Office, Role, Term, Type of Document.

More information about Pellucid generic and domain ontology is available in the [4] Description of Process Layer.

2.2. Framework for experience management

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, it is included the concept of Active Hint (AH), a representation of experience within the organization. Experience can be seen as knowing what to do in particular circumstances. The circumstances correspond to the context and the "know what to do" is characterized by the action and resources needed in that action.

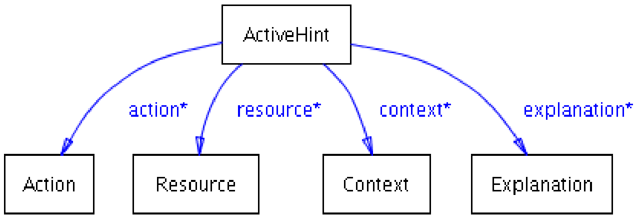

Figure 2. Active Hint concept

Pellucid has developed a framework for experience management [5] [6] in public organizations. The framework includes an analysis and design part as well as an experience management lifecycle (Fig. 3.). The analysis and design includes a contextual and conceptual modeling, where aspects of a generic public organization are modeled following the CommonKADS methodology. Pellucid has developed the concept of Active Hints, conveyors of experience regarded as suggestions to a user to perform some action that will assists his/her current activity. The final phase in the analysis and design is the functional design, where implementation details are specified taking into consideration all previous models. An important part in knowledge systems is the capture and dissemination of knowledge. The second element in the approach includes an experience management, where is explained how to capture and present experience as well as its evolution.

2.3. Experience management model

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 such components: analyzing employees' actions and workflow events; analyzing documents entered into the sys-tem; 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 organization. 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.

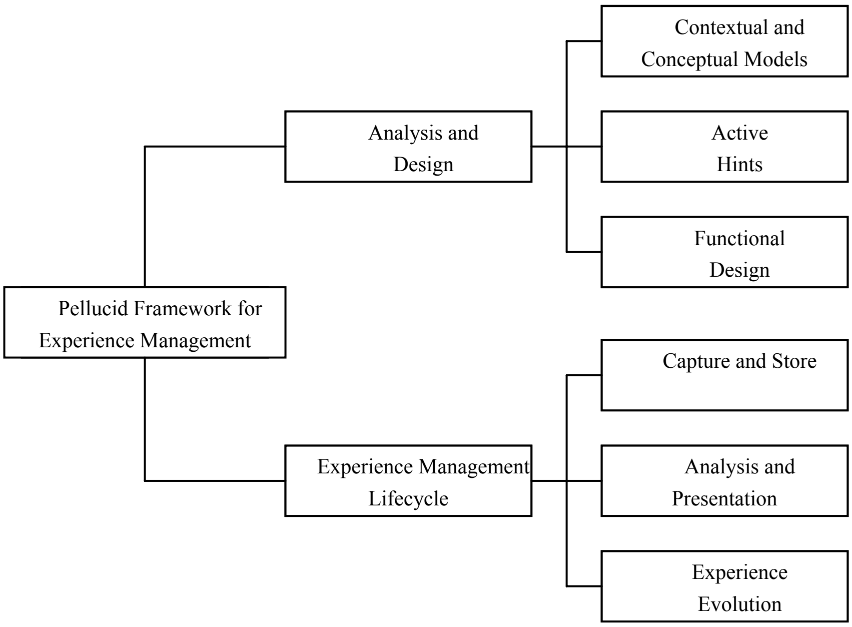
Pellucid Framework for Experience Management
Analysis and Design
Experience Management Lifecycle
Contextual and Conceptual Models
Active Hints
Functional Design
Capture and Store
Analysis and Presentation
Experience Evolution

Figure 3. Pellucid Framework for Experience Management

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 organization. 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 sys-tem. 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 aim of Experience Evolution phase is updating the available experience. The Pellucid system will include a set of methods and semi-automatic tools to allow knowledge engineers and expert users to update the experience stored in the Organizational Memory.

2.4. Related works

In the context of Pellucid, ontology is the main mechanism employed for the representation of information and knowledge, defining the meaning of the terms used in the agent content language and the relation among these terms. Similar methodology is also proposed by On-To-Knowledge that used Protégé as the tool for editing the ontologies in RDF(S)/OWL. The difference is the Pellucid platform is intended to serve the diverse needs of public organizations.

Among Experience Management projects, Decor and CoMMA have been the two most influential works on Pellucid. The way of knowledge delivery through active user support, triggered according to the context in a workflow, was also developed by Decor. The idea of active hints is originally conveyed in Decor, and is working it out in the Pellucid in a different direction as suggestions to the user to perform some actions that will assist her/his current activity. CoMMA defines an organizational memory using semantic-web technology and it is explored also in Pellucid, being the organizational memory one of the important components of the whole system.

More information about related works and state of art is available in the Pellucid Technical Report [7].

3. Functional design and platform architecture

The Pellucid platform consists of four main layers/components: Organizational Memory (OM), Process Layer, Interaction Layer and Information and Search Access (ISA) Layer.

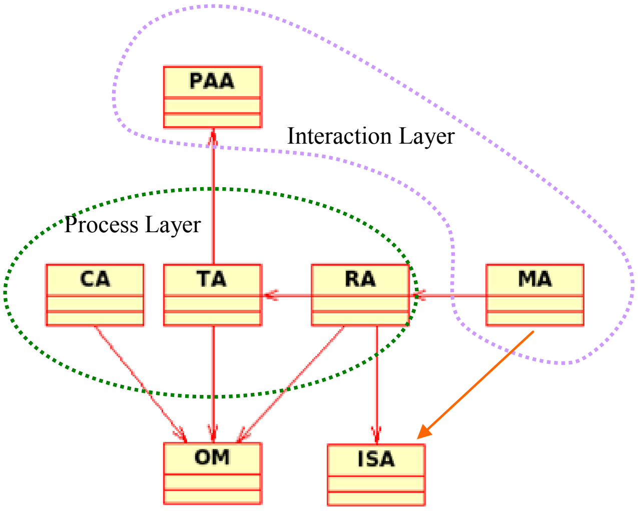

Figure 4. Pellucid platform architecture

3.1. Interaction Layer

The Interaction Layer comprises the Monitoring Agent (MA), which handle events incoming from the underlying Workflow Management/Tracking System (WfMS/WfTS) and monitoring sensors (piece of software plugged into repositories, developed according to the need of the Information Search Agent), and the Personal Assistant Agent, which capture user inputs and to present experiences to users in the way of Active Hints (AHs).

It also contains easy-customizable user-friendly Graphical User Interface (Pellucid GUI). The Interaction Layer of the Pellucid system encapsulates the interaction between Pellucid system, its main functionalities are: handling WfMS/WfTS events, capturing user inputs: adding notes, settings, and presenting active hints and related information/setting to users. The goal of the Interaction Layer is also design a solution that enables easy customization for pilot site applications and others possible applications in the future. Each pilot site has its own style and design suitable for its application. The customization of the Interaction Layer, therefore, must be compatible and comfortable with user-friendly design, easy to use and platform-independent. Therefore, HTML front-end technology with various middleware and server technology supports is chosen as the solution.

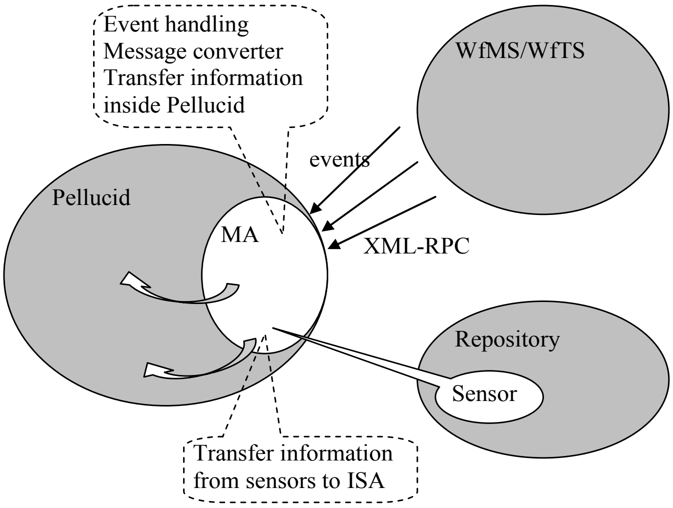

Figure 5. Input coming from external systems to the Pellucid platform

3.2. Process Layer

The core of the system is the Process Layer, which is in charge in generating experience - active hints - according to the current working context and expertise of the user includes Role Agent (RA), Task Agent (TA) and Capitalization Agent (CA). RA and TA are responsible for generating AHs from experiences stored in OM. CA deals with analyzing and creating new AHs by finding patterns and reusing experiences stored in OM. Process Layer works based on messages/events coming from Interaction Layer.

The realization of AH management includes determining working context (from capturing incoming information/events from MA and determining the working context in organization and updating current context of employees), handling Ahs (includes building, firing and sending AHs by RA and TA), creating new AHs from patterns found in the OM (main function of the CA).

There are AHs types produced in the Pellucid system:

- AHs from templates: when employee's context is changed by received event and the working context of AH templates available in the OM, AHs in OM become relevant for the employee and TA adds them into the employee's model.
- AHs templates are predefined hints in OM, entered during customization process, and/or by CA and during periodical knowledge maintenance.
- Resource-based AHs: AH templates available in the OM may contain RDQL queries built according to employee context. The re-sources (e.g. documents, contacts, etc.) are returned value of the evaluated patterns.
- Similarity-based AHs: similar resources are reported from ISA, and RA wraps it with an AH template. ISA detects any creation or modification of documents, emails and analyzes its content according to organization domain. Similarity meas-ures are defined by similarity function for each pilot application.
- AHs about critical activities: CA in background calculates and recommends the most suitable plan or notifies the critical activities (reduce the response time).

3.3. OM and its interaction with other layers

In the Organizational Memory (OM), important facts about the life, and history of the organization itself and its employees are stored. It can be considered as the working "mind" of an organization and plays a crucial role in experience management.

The ontology model is created using the Protégé2000. The creation process uses the ontology design done by academic partners and knowledge about pilot site applications from industrial partners with the help of the pilot site people. The output of the Protégé2000 is portable OWL file, which can be extended further according to users needs. It is mapped into OM through Jena interface at the first running (initial) time as ontology class definition with initial instances and property values (e.g. certain number of AH Templates). After that, when the Pellucid platform is running, events and resources are stored in OM by Pellucid agents through Jena interface. During the maintenance phase, the experienced experts update the class definition (e.g. number of AH templates can be increased: some intensive knowledge AHs will become AH templates).

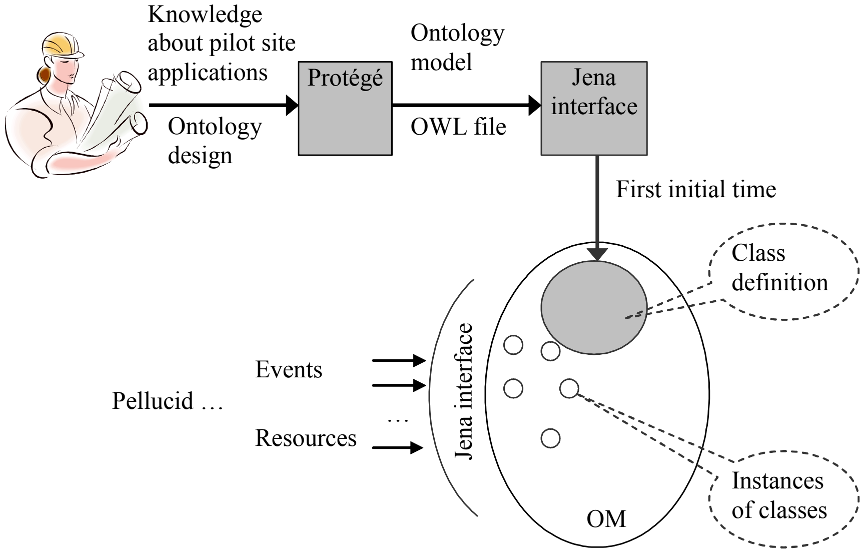

Figure 6. Ontology creation and usage in Pellucid – Interaction with OM

Monitoring Agent (MA), Role Agent (RA) and Task Agent (TA) work with a common ontology model that is maintained in OM. Capitalization Agent (CA) creates its own inference and loads data from OM to its local place. The Pellucid platform can exploit a model from databases or loaded directly from ontology files. Jena ontology model provides methods for ontology manipulation, inference upon the model and RDQL engine. The Information Search and Access (ISA) Layer provide the functionality for searching and retrieving resources from external repositories.

RDQL is a query language very similar to SQL for a relational database. The basic form of query is the SELECT. The most notably difference from SQL is that it misses the FROM keyword, because RDQL users should not be aware of the underlying storage mechanism. The basic form of a query is: SELECT <what> WHERE <condition>. The "what" condition is in the form of free-variables that will be assigned during query evaluation. The Jena library implements RDQL in a form optimized for the use with Relational Databases as the storage medium. It tries to convert the RDQL SELECT query in a relational SELECT on the underlying database.

4. Platform evaluation

The quality assessment was done firstly for the main components of the Pellucid platform making use of the standard metrics, e.g. Precision, Recall, F-measure, statistics measures and survey among pilot site users. Then the overall quality assessment of the Pellucid platform, as an experience management system, was done based on the final user evaluation [8], which gave the mark “very good” with “satisfied response time”. The general conclusion is that the users are happy with the framework and its scalability is acceptable for the selected kinds of usage. Further improvement would be profitable while moving toward real commercial implementation.

Although, the performance quality is not the main interest for the experience management system, it is a necessary and required to determine the response time of the platform under certain conditions. The main testing of behaviors quality is centered into the performance of the Process Layer (platform core). All evaluations are available in [9]. The evaluation shows different response time between the case when data is left in the database (MySQL) and all interface used Jena [10]with RDQL query languages [11] and the cache of loading data from database into local memory. The evaluation also shows that although Jena is the tools that provides manipulation with OWL/RDF, it performance with complicated RDQL queries is not good as expected. Other tools such as e.g. Joseki or Sesame, etc. were considered, their performances do not show as visibly better too. For Jena, its development is still in progress and it can evolve to be a leading tool with better and better properties.

The evaluation also specified the lower and higher bounds of performance of the Pellucid platform moreover some reassignment work among agent-based software component. Such evaluation is very important toward to industrial dissemination of the project.

5. Conclusion

The Pellucid system provides a generic solution of supporting system that works as an extension to underlying WfMS/WfTS. Agent-based components in the Process Layer cooperate with other Layers and Organizational Memory in order to provide the best assistance to employees. The Pellucid system find and generate useful active hints for each employee in every their working context. The Pellucid platform also provides an environment for experience management. It processes events reported from external systems, stores and analyzes them and in the future returns such experiences in the form of active hints appreciated for current situations. The recommendation from the Pellucid system is presented to end-users through the Interaction Layer with user-friendly and customizable GUI. Pellucid system has minimal requirements on automation of work-flow processes, and equipment of organization. A simple tracking system that is able to report events and track state of workflow processes is sufficient for Pellucid to support employees in executing their tasks.

This work is supported by the project Pellucid - "A Platform for Organizationally Mobile Public Employees", EU 5FP RTD IST-2001-34519 and Slovak national project VEGA No. 2/3132/23: "Effective tools and mechanisms for scaleable virtual organizations".

6. References

1. Pellucid project official homepage, http://www.sadiel.es/Europa/pellucid/.
2. A. Th. Schreiber, et al., *Knowledge Engineering and Management,* The CommonKADS methodology, ISBN 0-262-19300-0, The MIT Press, 2002.
3. OWL http://www.w3.org/TR/2004/REC-owl-semantics-20040210/
4. Description of Process Layer http://pellucid.ui.sav.sk/docs/Deliverable/D7%20(Final)%20Description%20of%20Process%20Layer.doc
5. A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutional Algorithms*, ISBN 3-540-42221-7, Springer-Verlag Berlin Heidelberg 2002.
6. R. Bergmann, *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, Lecture Notes in Artificial Intelligence, ISBN 3540441913, Springer-Verlag Telos, 2002.
7. Pellucid Technical Report, May 2004, http://pellucid.ui.sav.sk/docs/TechnicalReports/TechnicalReportPellucidMay21.doc.
8. Pellucid deliverable D11 – Final User Evaluation and Best Practice Guidelines (pre-final version) http://pellucid.ui.sav.sk/docs/Deliverable/D11%20(Pre-Final)%20Final%20User%20Evaluation%20and%20Best%20Practice%20Guidelines.pdf.
9. Pellucid deliverable DX1 - Quality and performance assessment of the Pellucid platform (pre-final version) http://pellucid.ui.sav.sk/docs/Deliverable/Pellucid%20DX1%20(Pre-Final).pdf.
10. Jena – A Semantic Web Framework for Java http://jena.sourceforge.net.
11. W3C - RDQL - RDF Data Query Language, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/.
12. T.G. Nguyen, M. Laclavik, Z. Balogh, E. Gatial, L. Hluchy, T.T. Dang, I. Budinska, *Pellucid - Platform for Organisational Public Emploees,* ICETA'2004, pp. 16-18, IEEE Computer Society, ISBN 80-89066-85-2, Kosice, Slovakia, September 2004.
13. M. Laclavik, Z. Balogh, L. Hluchy, G.T.Nguyen, I. Budinska, T.T. Dang, *Pellucid Agent Architecture for Administration Based Processes*, IAWTIC 2003, pp. 577-584, ISBN 1740880692, CD-ROM, Vienna, Austria, February 2003.
14. J. Ferber, *Multi-agent system: An introduction to distributed Artificial intelligence*, ISBN 0-201-36048-9, Addison Wesley Longman, 1999.