AGENT-BASED DOCUMENT MANAGEMENT WITHIN THE WHOLE LIFECYCLE OF INFORMATION SYSTEM

Viacheslav Lanin, George Sokolov

Abstract*: In this paper an approach to using unstructured documents intelligent processing at different stages of information system's lifecycle is described. When documents are processed it's suggested to use the results of semantic indexing. Every document is mapped to one agent that "represents the interests" of the document. This approach allows solving a wide range of problems which are appeared not only with the information system functioning but also system designing.*

Keywords*: semantic indexing; agent; ontology; document.*

ACM Classification Keywords*: I. Computing Methodologies. I.2 ARTIFICIAL INTELLIGENCE: I.2.11 Distributed Artificial Intelligence – Intelligent agents; Multiagent systems. I.7 DOCUMENT AND TEXT PROCESSING: I.7.2 Document Preparation – Index generation.*

Introduction

Electronic documents processing is one of the main functions of every information system. Generally documents have a passive role. The documents are only containers for some kinds of information. However if the activity is added to documents, it'll be able to optimize a solution to many traditional problems of information systems and also to solve some new problems. In this paper it's assumed to solve the problem by using the agent paradigm. Namely for giving "dynamics" of documents specialized intelligent agents that the represent interests of the document will be used.

It's needed to solve a range of problems for implementation of the proposed approach. These problems are the following:

- document semantic indexing (representation);
- creating/choosing/deploying of an agent platform (describing language, communications);
- developing of information system architecture according to suggested approach.

Related works

The problem of adding to electronic documents data processing logic has been solving for many years. There are commercial as well as research developments. Among the commercial solutions one can highlight the following approaches. They are

- office programming tools and macro programming languages;
- adding context-sensitive logic to documents;
- solutions used for automation of electronic forms filling (Microsoft InfoPath, Google Forms).

The office programming assumes the development of applications that are designed for automation of office activity with the use of a specialized batch (Visual Basic for Application в Microsoft Office, LibreOffice Basic) [Krieger, 2009]. In the office environment a program project is linked with a document inseparably. The project is stored as a part of the document and can't exist independently. Thus the goal of development is document, but not a program. Practice area of office programming is very wide. It's from setting of separate documents to problem-solving of office activity's automation. Office development tools are focused on the users, not on the professional programmers. For example Microsoft Office gives two options of program creating. They are using of MacroRecorder and handle coding (using VBA programming language). MacroRecorder is a program tool that records user's actions when he works with documents and applications. This tool saves user's actions as VBA source code. Currently Microsoft plans to replace VBA tools with VSTO specialized tool [Fox, 2008] and modern script languages [Oliver, 2013].

Microsoft Smart Documents are a programmability feature of Microsoft Word and Excel [Kelly, 2003]. Developers can create Word or Excel Smart Documents augmented with programmable pieces that help users create or modify them. Reusing boilerplate text, implementing business rules for how a document should be formatted or completed, and routing a document through an approval process are all procedures that Office Smart Documents can simplify. Smart Documents and associated controls are installed when the user simply opens the document obtained from a trusted server or an attachment in an e-mail message. They can automatically update themselves from a trusted server, greatly simplifying update deployment.

In research projects an agent approach becomes popular [Ginsburg, 1999; Ginsburg, 2000; Pešović, 2011]. In most solutions one assumes to use agents for workflow management and document lifecycle.

One of the most interesting research projects is an approach described in [Dourish, 2000a; Dourish, 2000b]. Authors say that document properties are a compelling infrastructure on which to develop document management applications. Assumed paradigm was implemented in the system which was called «Placeless Documents». According to the assumed approach user-specific and active properties are added to the document. A property-based approach avoids many problems of the traditional hierarchical storage mechanisms, reflects document organizations meaningful to user tasks, provides a means to integrate the perspectives of multiple individuals and groups, and does this all within a uniform interaction framework. Document properties can reflect not only categorizations of documents and document use, but also expressions of desired system activity, such as sharing criteria, replication management and versioning. Augmenting property-based document management systems with active properties that carry executable code enables the provision of document-based services on a property infrastructure. The combination of document properties as a uniform mechanism for document management, and active properties as a way of delivering document services, represents a new paradigm for document management infrastructures.

Table 1. Generic agent-based platforms

	JADE [Bellifemine, 2010]	**MASDK** [Gorodetsky, 2009]	**ZEUS** [Collis, 1999]
Developer community	Telecom Italia Lab	SPIIRAS	BT Laboratories
License	LGPL	LGPL	LGPL
Description	This is the platform for rapid development of multi-agent systems, which implements FIPA standards. JADE provides base classes for creating agents and infrastructure for the operation of multi-agent system.	This is the software environment for multi-agent application development that supports the full life cycle application development of MAS. The agent platform, which is the part of MASDK, works on the principle of P2P.	This is the agent platform designed for rapid development of multi-agent applications. Zeus provides a library of agent components.
Description of the agent behavior	Set in the code of the agent class that inherits from Agent.	Set with language ASML. This language is used for generating applied MAS.	Set in an environment for building agents, from which the agent code is generated.

Further different agent platforms for agent architecture implementing will be considered.

The most popular agent platforms are JADE [1], MASDK [2], Zeus [3] (see table 1).

Each of these agent platforms allows one way or another to describe the behavior of the agent. Depending on the platform we can define almost any behavior of an agent, programming or describing it using specific language. So we can determine the behavior of the agent that implements mechanisms of semantic indexing and interest's representation of each document. Authors think that the most appropriate agent platform for agent architecture implementing is JADE. JADE has the most appropriate architecture for solvable problem (FIPA abstract architecture of agent platform) and also has rich documentation, worldwide community.

Approach description

The main idea of the approach is that every document in the system (even potential) has an agent which is linked with it. The agent owns full information about the document and can represent the interest of the document in a wide range of problems solving. For a document management it's needed for the agent which represents document's interests to own knowledge about document's semantic and possess to ability to interpret information that is included in it. For this it's necessary to solve the problem of semantic indexing. In [Bessonov, 2012] the approach to the solution to this problem is suggested. Besides the approach to inclusion of created semantic index in a document body is described in this paper. Also it's suggested that the problems of domain's concepts are described in ontology. The ontology is available for agents and can be interpreted by agents.

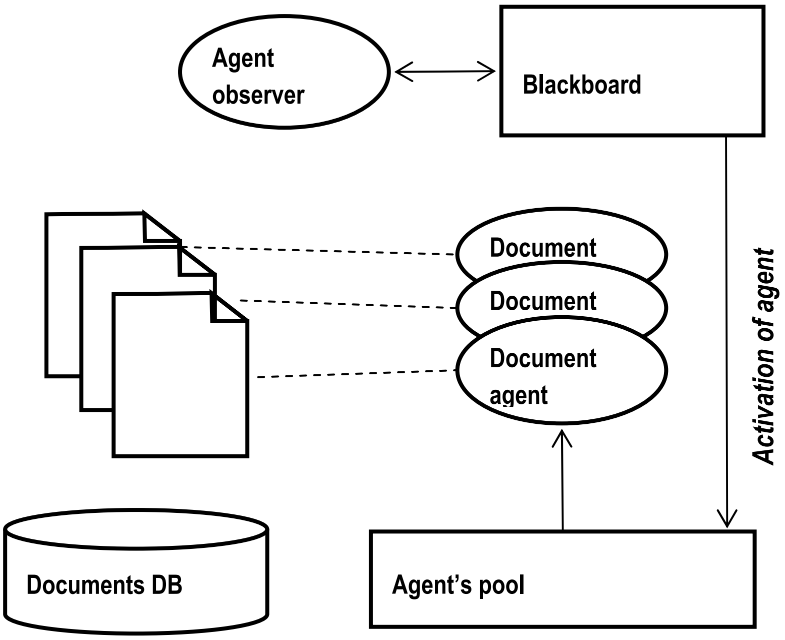


Figure 1. System architecture

Documents are stored in a specialized database together with their semantic indexes. When some operations on the document are invoked (creating, browsing, updating, deleting) the document agent is activated. It controls and manages operation's performing. All agents that are registered in the system are contained in the agent's pool. The agent's pool is a storage of inactive agents. Information about the agent competence (knowledge about document) and subscriptions to obtaining information is stored in the blackboard. If necessary (load's raising or appearing of new task) additional instances of agents can be created. For agent's activation specialized agent- observer is used. It monitors current working context and sends a message to concrete document agent if necessary. Also it could stay active autonomously and perform monitoring.

Problems that are solved within this approach

Based on the described approach it's planned to automate tasks of document lifecycle's support in functioning of information system and support of information system's development, analysis of problem domain.

- Messages about changing of document's content can be used in updating entity's attributes that information about them is stored in information system. Since document has a semantic index agent document using problem domain's ontology can determine changed objects and can alert the system about this.
- Messages about document's data changing can be used for automatic actualization of documents.
- Information about document's template changing also allows updating document timely.
- Based on semantic analysis of documents a set of potential entities and their attributes can be obtained in system's designing.
- When information system is designed it can be performed a search of dependencies and establishing links between documents that regulate an activity of business-system. Result of document's analysis is automatic ontology building. Nodes of this ontology are analyzed documents and their concepts. So as the result of analysis of the system of interrelated documents must be built. Key points of this system are given below.
  a) the system of documents is related to some activity's directions of business-system (related to concepts, objects of problem domain);
  b) the system of documents reflects links between these concepts (document or set of documents could be linked with every concept, links between documents reflect links between concepts);
  c) the system of documents contains regulatory information that could be obtained based on analysis of document's content.
- Analysis of project's documentation allows highlighting key dates of project and project's report documents.

Conclusion

The approach described in this paper allows solving a problem of adding its processing logic to the document. Unlike another approaches source code isn't added to the document and no attribute information is assigned. This approach allows abstracting from technologic features and specifics of concrete document's formats. In turn used agent approach promotes rising of system's adaptability and its intellectuality.

Suggested approach allows solving a wide range of problems that are connected with designing of information system and its further functioning. When it's focused on documents handling one could say about document-oriented paradigm of information system lifecycle's support.

Acknowledgement

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Bellifemine, 2010] F. Bellifemine, Giovanni Caire, Tiziana Trucco, Giovanni Rimassa JADE Programmers guide. 2010. Available: http://jade.tilab.com/doc/programmersguide.pdf.

[Bessonov, 2012] V. Bessonov, V. Lanin, G. Sokolov A semantic indexing of electronic documents in open formats International Journal “Information Theories and Applications”, Vol. 19, Number 2, 2012, p. 139-148.

[Collis, 1999] J. Collis, D. Ndumu ZEUS Techical Manual. 1999. Available: www.upv.es/sma/plataformas/zeus/Zeus-TechManual.pdf.

[Dourish, 2000a] P. Dourish, K. Edwards, A. LaMarca, J. Lamping, K. Petersen, M. Salisbury, Douglas B. Terry, J. Thornton Extending Document Management Systems with User-Specific Active Properties, in ACM Transactions on Information Systems, vol. 18, no. 2, ACM, New York, USA, 2000, pp. 140–170.

[Dourish, 2000b] P. Dourish, K. Edwards, J. Howell, A. LaMarca, J. Lamping, K. Petersen, M. Salisbury, D. Terry, J. Thornton A Programming Model for Active Documents, in UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, New York, USA, 2000, pp. 41-50.

[Fox, 2008] S. Fox Developing Office Business Apps with Visual Studio 2008 MSDN Magazine Launch. 2008. Available: http://msdn.microsoft.com/en-us/magazine/cc164242.aspx.

[Ginsburg, 1999] M. Ginsburg An Agent Framework for Intranet Document Management // Autonomous Agents and Multi-Agent Systems, Volume 2, Issue 3, 1999, pp 271-286.

[Ginsburg, 2000] M. Ginsburg Intranet Document Management Systems as Knowledge Ecologies. Proceedings of the 33rd Hawaii International Conference on System Sciences. Volume 3 (HICSS '00), Vol. 3. IEEE Computer Society, Washington, USA, 2000, p. 3017.

[Gorodetsky, 2009] Gorodetsky V., Karsan O., Samoilov V., Serebryakov S. Applied multi-agent systems of group control // Scientific and Technical Information Processing, December 2010, Volume 37, Issue 5, pp 301-317.

[Kelly, 2003] M. Kelly Create Word and Excel Smart Documents with C++ and XML, MSDN Magazine, 2003, December. Available: http://msdn.microsoft.com/en-us/magazine/cc164042.aspx.

[Krieger, 2009] S. Krieger Using VBA to Access Document Content in the 2007 Office System, March, 2009 Available: http://msdn.microsoft.com/en-us/library/office/dd630624(v=office.12).aspx.

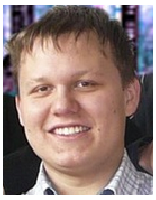
[Oliver, 2013] S. Oliver, E. Schmidt Exploring the JavaScript API for Office: Data Binding and Custom XML Parts. MSDN Magazine. 2013, VOL. 28, # 4. Available: http://msdn.microsoft.com/en-us/magazine/dn166930.aspx

[Pešović, 2011] D. Pešović, M. Vidaković, M. Ivanović, Z. Budimac, J. Vidaković Usage of agents in document management. ComSIS Vol. 8, No. 1, 2011, Novi Sad, Serbia, pp. 193-210.

Authors' Information

Viacheslav Lanin *– National Research University Higher School of Economics, Department of Business Informatics; Russia, Perm, 614070, Studencheskaya st., 38; e-mail: lanin@perm.ru.*

Major Fields of Scientific Research: Intelligent agents, Ontologies, Document processing.



George Sokolov *– Perm State National Research University, Computer Science Department; Russia, Perm, 614990, Bukireva St.,15; e-mail: sokolovgeorge@gmail.com.*

Major Fields of Scientific Research: Intelligent agents, Semantic indexing, Graph drawing.