

Boston, Massachusetts
July 9-11, 2007

2007



PROCEEDINGS

SEKE 2007

**The 19th International Conference on
Software Engineering &
Knowledge Engineering**

Sponsored by

Knowledge Systems Institute Graduate School, USA

Technical Program

July 9-11, 2007

Hyatt Harborside Hotel, Boston, Massachusetts, USA

Organized by

Knowledge Systems Institute Graduate School

Copyright © 2007 by Knowledge Systems Institute Graduate School

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher.

ISBN 1-891706-20-9 (paper)

Additional Copies can be ordered from:
Knowledge Systems Institute Graduate School
3420 Main Street
Skokie, IL 60076, USA
Tel: +1-847-679-3135
Fax: +1-847-679-3166
Email: office@ksi.edu
<http://www.ksi.edu>

Proceedings preparation, editing and printing are sponsored by
Knowledge Systems Institute Graduate School

Printed by Knowledge Systems Institute Graduate School

Table of Contents

Foreword	iii
Conference Organization	iv
Keynote	1
<i>Dr. Joe Urban</i>	
Software Engineering Methodology	
Constructing Self-Adaptive Systems with Polymorphic Software Architecture <i>Xiaoxing Ma, Yu Zhou, Jian Pan, Ping Yu, Jian Lu</i>	2
Odyssey-MDA: A transformational approach to component models <i>Natanael E. N. Maia, Ana Paula Terra Bacelo, Claudia M. Werner</i>	9
An Empirical Expository Study on Inferring Developers' Activities from Low-Level Data <i>Irina Coman, Alberto Sillitti (S)</i>	15
TRAP.NET: A Realization of Transparent Shaping in .NET <i>S. Masoud Sadjadi, Fernando Trigos</i>	19
A Framework for Selecting Agile Practices and Defining Agile Software Processes <i>Patricia Vilain, Priscila Basto Fagundes, Thiago Leao Machado (S)</i>	25
A Proposal to Delegate GUI Implementation using a Source Code based Model <i>Marco Monteiro, Paula Oliveira, Ramiro Goncalves (S)</i>	29
Cost-based Analysis of Multiple Counter-Examples <i>Flavian Vasile, Samik Basu</i>	33
Common Coupling as a Measure of Reuse Effort in Kernel-Based Software <i>Liguo Yu, Stephen R. Schach, Kai Chen</i>	39
An Approach to Validating Translation Correctness From SAM to Java <i>Yujian Fu, Zhijiang Dong, Gonzalo Argote-Garcia, Leyuan Shi, Xudong He (S)</i>	45

Software Processes and Engineering Practice

QSEE Project: An Experience in Outsourcing Software Development for Space Applications <i>Valdivino Santiago, Fatima Mattiello-Francisco, Ricardo Costa, Wendell Pereira da Silva, Ana Maria Ambrosio</i>	51
Broadening the Use of Process Patterns for Modeling Processes <i>Hanh Nhi Tran, Bernard Coulette, Bich Thuy Dong</i>	57
A Framework for Tailoring Software Process <i>Lisandra M. Fontoura, Roberto T. Price (S)</i>	63
Analyzing Configuration Management Repository Data for Software Process Improvement <i>Shihong Huang, Christopher Lo (S)</i>	67

Aspect-Based Software Development

Smooth Quality Oriented Component Integration through Product Line Based Aspect-Oriented Component Adaptation <i>Yankui Feng, Xiaodong Liu, Jon Kerridge</i>	71
Modular Specification of Aspect-oriented Systems and Aspect Conflicts Detection <i>Hui Liang, Jing Sun (S)</i>	77
Avoiding Bad Smells in Aspect-Oriented Software <i>Eduardo K. Piveta, Marcelo Hecht, Ana Moreira, Marcelo S. Pimenta, Joao Araujo, Pedro Guerreiro, R. Tom Price</i>	81

Software Testing and Quality Assurance

Metrics of Credibility and Interaction Quality: Design and Evaluation <i>Nilda Perez Otero, Marcelo Perez Ibarra, Sandra Mendez, Adelina Garcia, Ma. del Pilar Galvez Diaz, Viviana E. Quincoces, Hector Liberatori, Beatriz Fiorito, Cecilia Maria Lasserre</i>	87
Predicting Order of Likelihood of Defective Software Modules <i>Rattikorn Hewett, Phongphun Kijsanayothin, Alta van der Merwe</i>	93
Automated Test Code Generation from UML Protocol State Machines <i>Dianxiang Xu, Weifeng Xu, W. Eric Wong</i>	99

Validating A Layered Decision Framework for Cost-Effective Network Defense <i>Huaqiang Wei, Jim Alves-Foss, Du Zhang</i>	105
Toward Modeling and Analysis for Software Installation Testing <i>Jerry Gao, Sujana Tirumalasetti, Chien-Pin Hsu, Yip Cheong, Anne Colendich, Todd Fitch (S)</i>	111
Automatic Test Generation for Database-Driven Applications <i>Zhenyu Dai, Mei-Hwa Chen</i>	117
Fault-Based Testing of Data Schemas <i>Maria Claudia F. P. Emer, Silvia Regina Vergilio, Mario Jino</i>	123
NLForSpec: Translating Natural Language Descriptions into Formal Test Case Specifications <i>Daniel Leitao, Dante Torres, Flavia Barros</i>	129
Enhanced Random Testing for Programs with High Dimensional Input Domains <i>F.-C. Kuo, K.-Y. Sim, Chang-ai Sun, S.-F. Tang, Zhi Quan Zhou</i>	135
On Test Case Distributions of Adaptive Random Testing <i>Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu (S)</i>	141
Reducing the Number of Test Cases for Performance Evaluation of Components <i>Joco W. Cangussu, Kendra Cooper, Eric Wong</i>	145
Combining Decorated Classification Trees with RCPS Stochastic Models to Gain New Valuable Insights into Software Project Management <i>Antonio Juarez Alencar, Gelson Guedes Rodrigues, Eber Assis Schmitz, Armando Leite Ferreira</i>	151
Towards a Reference Architecture for Software Testing Tools <i>Elisa Yumi Nakagawa, Adenilso da Silva Simao, Fabiano Ferrari, Jose Carlos Maldonado</i>	157
Controlling Restricted Random Testing: An Examination of the Exclusion Ratio Parameter <i>Kwok Ping Chan, T.Y. Chen, Dave Towey (S)</i>	163
An Approach to Software Testing of Machine Learning Applications <i>Christian Murphy, Gail Kaiser, Marta Arias (S)</i>	167

System Requirements Analysis, Modeling and Specification

Agile Methods and Quality Models: Towards an Integration in Requirements Engineering	173
<i>Alexandre Lazaretti Zanatta, Patricia Vilain</i>	
A Framework of Hierarchical Requirements Patterns for Specifying Systems of Interconnected Simulink/Stateflow Modules	179
<i>Changyan Zhou, Ratnesh Kumar, Devesh Bhatt, Kirk Schloegel, Darren Cofer</i>	
In the Requirements Lies the Power	185
<i>Rand Waltzman, Kristina Winbladh, Thomas A. Alspaugh, Debra J. Richardson</i>	
Data and Process Analyses of Data Warehouse Requirements	191
<i>Estella Annoni, Franck Ravat, Olivier Teste</i>	
Requirement Analysis Evolution through Patterns	197
<i>Luca Vetti Tagliati, Roger Johnson, George Rousso (S)</i>	
Automatic Generation of Use Case Diagrams from English Specifications Document	203
<i>Nathalie Rose T. Lim, Christobal T. Cayaba, Joseph Astrophel E. Rodil</i>	
REM4j - A framework for measuring the reverse engineering capability of UML CASE tools	209
<i>Steven Kearney, James F. Power</i>	
FlexUML: A UML Profile for Flexible Process Modeling	215
<i>Ricardo Martinho, Dulce Domingos, Joao Varajao</i>	
A Formal Specification for Product Configuration in Software Product Lines	221
<i>Huilin Ye, Yuqing Lin</i>	
Designing a Platform-Independent Use-Case for a Composite Application using a Reference Architecture	227
<i>Helge Hofmeister, Guido Wirtz (S)</i>	
Synchronization of UML Based Refactoring with Graph Transformation	232
<i>Y. Kosker, A. B. Bener</i>	
Using Formal Composition of Use Cases in Requirements Engineering	238
<i>Rabeb Mizouni, Aziz Salah, Rachida Dssouli</i>	

Web-Based Applications

Real-Time Trust Management in Agent Based Online Auction Systems <i>Rinkesh Patel, Haiping Xu, Ankit Goel</i>	244
Geometric Thumbnails For Web Searching <i>Chris Dunn, Beomjin Kim (S)</i>	251

Web Technology and Web Engineering

ADKwik: Web 2.0 Collaboration System for Architectural Decision Engineering <i>Nelly Schuster, Olaf Zimmermann, Cesare Pautasso</i>	255
Improving Usability of Web Systems with Similar Business Objectives <i>Rashid Ahmad, Zhang Li, Farooque Azam</i>	261
Processing Manipulations of Context Information on the Web <i>Roberto De Virgilio</i>	268
A Tag-Level Web-Caching Scheme for Reducing Redundant Data Transfers <i>Steven E. Cox, Du Zhang, Jinsong Ouyang</i>	274

Software Project and Resource Management

Methodology for Reusing Human Resources Management Standards <i>Asuncion Gomez-Perez, Jaime Ramirez, Boris Villazon-Terrazas</i>	280
An Adaptive Resource Management Approach for a Healthcare System <i>Claudia Raibulet, Luigi Ubezio, Stefano Mussino</i>	286
Study of the Relationships between Personality, Satisfaction and Product Quality in Software Development Teams <i>Marta Gomez, Silvia T. Acuna (S)</i>	292

Software Reuse and Component Technology

Towards Constructing High-available Decentralized Systems via Self-adaptive Components <i>Xi Sun, Li Zhou, Lei Zhuang, Wenpin Jiao, Hong Mei</i>	296
---	-----

SAFES: A Static Analysis for Field Security in Java Components <i>Aiyu Shi, Gleb Naumovich</i>	302
Reuse of Database Access Layer Components in JEE Product Lines: Limitations and a Possible Solution (Case Study) <i>Ding Peng, Stan Jarzabek, Damith C. Rajapakse, Hongyu Zhang</i>	308
Design of Wrapper for Self-Management of COTS Components <i>Michael E. Shin, Fernando Paniagua</i>	314
QoS-Optimized Integration of Embedded Software Components with Multiple Modes of Execution <i>Zonghua Gu, Qingxu Deng</i>	320
A C++ Framework for Developing Component Based Software Supporting Dynamic Unanticipated Evolution <i>Andre Rodrigues, Hyggo Almeida, Angelo Perkusich</i>	326
Representing Design Rationale to support Reuse <i>Adriana Pereira de Medeiros, Daniel Schwabe (S)</i>	332
Software System Maintenance	
Telling Stories about System Use: Capturing Collective Tacit Knowledge for System Maintenance <i>Adriana Cristina de Oliveira, Renata Mendes de Araujo, Marcos R.S. Borges</i>	337
Evolution and Runtime Monitoring of Software Systems <i>Hui Liang, Jin Song Dong, Jing Sun</i>	343
On Modern Debugging For Rule-Based Systems <i>Valentin Zacharias, Andreas Abecker</i>	349
Truth Eliciting Mechanisms for Trouble Ticket Allocation in Software Maintenance Services <i>Karthik Subbian, Y. Narahari</i>	355
Knowledge Engineering, Natural Language Processing, and AI	
Graphical Notation for Natural Language and Knowledge Representation <i>Magda G. Ilieva</i>	361

A Hybrid Approach for Natural Language Query Translation <i>Pornpimon Teekayuphun, Ohm Sornil</i>	368
Effective Fault Localization using BP Neural Networks <i>W. Eric Wong, Lei Zhao, Yu Qi, Kai-Yuan Cai, Jing Dong</i>	374
Temporal Software Change Prediction Using Neural Networks <i>Mehdi Amoui, Mazeriar Salehie, Ladan Tahvildari</i>	380
Do Neural-Network Question-Answering Systems Have a Role to Play in the Deployment of Real World Information Systems? <i>Antonio Juarez Alencar, Renata Chaomey Wo, Eber Assis Schmitz, Armando Leite Ferreira</i>	386
Knowledge Conversion in Software Development <i>Olivier Gendreau, Pierre N. Robillard</i>	392
A Language Facilitating Informal Reasoning about Programs <i>J. Nelson Rushton, Dwayne Towell</i>	396

Plenary Talk

Towards Seamless Business Process and Dialogue Specification <i>Dr. Dirk Draheim</i>	402
---	-----

Database Retrieval Methods

Evaluating the Efficiency of Retrieval Methods for Component Repositories <i>Oliver Hummel, Werner Janjic, Colin Atkinson</i>	404
Benchmarking the RDF(S) Interoperability of Ontology Tools <i>Raul Garcia-Castro, Asuncion Gomez-Perez, York Sure</i>	410
A Deep Classification of Temporal Versioned Integrity Constraints for Designing Database Applications <i>Robson Leonardo Ferreira Cordeiro, Renata de Matos Galante, Nina Edelweiss, Clesio Saraiva dos Santos (S)</i>	416
Generating Linear Temporal Logic Formulas for Pattern-Based Specifications <i>Salamah Salamah, Vladik Kreinovich, Ann Q. Gates (S)</i>	422

Ontology Based Classification Generating Method for Browsing-Based Component Retrieval	428
<i>Ge Li, Lu Zhang, Bing Xie, Weizhong Shao (S)</i>	

Data

A Context-Dependent Semantic Distance Measure	432
<i>Ahmad El Sayed, Hakim Hacid, Djamel Zighed</i>	
A Semantical Change Detection Algorithm for XML	438
<i>Rodrigo Cordeiro dos Santos, Carmem Hara</i>	
XML Schema Evolution by Context Free Grammar Inference	444
<i>Julio C. T. da Silva, Martin A. Musicante, Aurora T. R. Pozo, Silvia R. Vergilio</i>	

Software Development and Design Pattern

Software Tradeoff Assistant: An Integrated Framework for Analytical Decision Making and Tradeoffs in Software Development	450
<i>Ratikorn Hewett, Vikram Patankar</i>	
Improving Separation of Concerns in the Development of Scientific Applications	456
<i>S. M. Sadjadi, J. Martinez, T. Soldo, L. Atencio, R. M. Badia, J. Ejarque (S)</i>	
Pattern-based J2EE Application Deployment with Cost Analysis	462
<i>Nyun Zhang, Gang Huang, Ling Lan, Hong Mei (S)</i>	
Exploratory Design of Derivation Business Rules Using Query Rewriting	467
<i>Roman Krenicky, David Willmor, Suzanne M. Embury (S)</i>	
Classification of Design Pattern Traits	473
<i>Jing Dong, Yajing Zhao (S)</i>	

Data Warehouse

A Proposal for a Conceptual Data Warehouse Quality Model	477
<i>Manuel Serrano, Rafael Romero, Jose-Norberto Mazon, Juan Trujillo, Mario Piattini</i>	
Integrating Complex Data into a Data Warehouse	483
<i>F. Rava, O. Teste, R. Tournier, G. Zurfluh (S)</i>	

Data Mining and Machine Learning

- Learning from Software Quality Data with Class Imbalance and Noise
Andres Folleco, Taghi M. Khoshgoftaar, Jason Van Hulse, Chris Seiffert 487

System and Software Architecture

- Architectural Elements Recovery and Quality Evaluation to Assist in Reference Architectures Specification
Aline Pires Vieira de Vasconcelos, Claudia Maria Lima Werner 494
- EvoSpaces: 3D Visualization of Software Architecture
Sazzadul Alam, Philippe Dugerdil 500
- Ontobrowse: A Semantic Wiki for Sharing Knowledge about Software Architectures
Hans-Jorg Happel, Stefan Seedorf 506

Applications

- Building Business Considerations into Enterprise Application Designs
Rattikorn Hewett, Aashay Thipse 513
- Incremental effort prediction models in Agile Development using Radial Basis Functions
Raimund Moser, Witold Pedrycz, Giancarlo Succi (S) 519
- BASS: Business Application Support through Software Services
Mateus B. Costa, Rodolfo F. Resende, Marcelo V. Segatto, Eduardo F. Nakamura, Nahur Fonseca 523

Model-Driven Software Development

- Using Model-Driven Pattern Matching to derive functionalities in Models
Ignacio Garcia-Rodriguez de Guzman, Macario Polo, Mario Platini 529
- A Model-driven Approach to Architecting Secure Software
Ebenezer A. Oladimeji, Sam Supakkul, Lawrence Chung (S) 535

Agent-Based Technology and Intelligence

An Intelligent Agent of Automatically Notify Services <i>Shuo-Yan Hsu, William C. Chu</i>	541
A Proposal for a Decentralized Multi-Agent Architecture for Virtual Enterprises <i>Andreas Grunert, Sven Kaffille, Guido Wirtz</i>	546
Traceability for Agent-Oriented Design Models and Code <i>Gilberto Cysneiros, Andrea Zisman</i>	552
ONTOMADEM: An Ontology-driven Tool for Multi-Agent Domain Engineering <i>Rosario Girardi, Adriana Leite</i>	559
A Three Level Multi-agent Architecture to Foster Knowledge Exchange <i>Juan Pablo Soto, Aurora Vizzaino, Javier Portillo-Rodriguez, Mario Piattini (S)</i>	565
An Agent Based System for Search in Distributed Environments <i>Li Sa, Yong-Sheng Ding (S)</i>	570

DB Access and Query Processing

Tree Hash Under Concurrency Control <i>Kyosuke Yasuda, Takao Miura</i>	574
Query Processing in Paraconsistent Databases in the Presence of Integrity Constraints <i>Navin Viswanath, Rajshekhar Sunderraman</i>	580
An Object-Oriented Approach to Storage and Retrieval of RDF/XML Documents <i>Ching-Ming Chao</i>	586
CXPath: a Query Language for Conceptual Models of Integrated XML Data <i>Diego de Vargas Feijo, Claudio Naoto Fuzitaki, A Ivaro Moreira, Renata de Matos Galante, Carlos Alberto Heuser</i>	592

Service-Oriented Technology and Web Technology

OWLed: Extending Knowledge for Web Ontology Language <i>Hichem Zait, Aicha Mokhtari</i>	598
Using Ontologies to Represent Software Project Management Antipatterns <i>Dimitrios Settas, Ioannis Stamelos</i>	604

Service Composition Using Planning and Case-Based Reasoning <i>Kuan-Hsian Huang, Alan Liu</i>	610
MDA-based Ontology Development: A Study Case <i>Eluzai Souza dos Santos, Celia Ghedini Ralha, Hervaldo Sampaio Carvalho</i>	616
Towards Domain-Centric Ontology Development and Maintenance Frameworks <i>Faezeh Ensan, Weichang Du (S)</i>	622
Service Oriented Architecture Empirical Study <i>Mohammad Abu-Matar, Jeff Offutt (S)</i>	628
Semantic Support to Reformulate Public Services in Terms of Life Events <i>Luis Alvarez Sabucedo, Luis Anido Rifon (S)</i>	632
A Component-Based Solution and Architecture for Dynamic Service-Based Applications <i>Alessio Colzi, Tommaso Martini, Paolo Nesi, Davide Rogai (S)</i>	637
System Reliability and Verification	
Adequacy of Composite Parametric Software Reliability Models <i>Lance Fiondella, Swapna S. Gokhale</i>	643
Evaluation of the OORT Techniques for Inspection of Requirements Specifications in UML: an empirical study <i>Tereza G. Kirner, Erik R. da Cruz (S)</i>	649
Agent Modeling/Methodology	
Adjudicator: A Statistical Approach for Learning Ontology Concepts from Peer Agents <i>Behrouz Far, Abdel-Halim Hafez Elamy, Nora Houari, Mohsen Afsharchi</i>	654
Agent Applications	
DALICA: Intelligent Agents for User Profile Deduction <i>Stefania Costantini, Leonardo Mostarda, Arianna Tocchio, Panagiota Tsintza (S)</i>	660

Human Interaction and GUI Development

- An Approach to Multimodal Input Interpretation in Human-Computer Interaction
Fernando Ferri, Patrizia Grifoni, Stefano Paolozzi 664
- Sketch Style Recognition in Human Computer Interaction
Daniilo Avola, Fernando Ferri, Patrizia Grifoni 670
- Human-Computer Interaction for a Novel Arm-wrestling Robot
Chul-goo Kang, Ho-yeon Kim (S) 676

Service

- Managing XML Versions and Replicas in a P2P Context
Deise de Brum Saccol, Nina Edelweiss, Renata de Matos Galante, Carlo Zaniolo 680
- Knowledge sharing through a simple release planning method for web application development
Sven Ziemer, Ilaria Canova Calori 686
- Distributed BPEL Processes
Luciano Baresi, Andrea Maurino, Stefano Modafferi 692
- SAM: Semantic Advanced Matchmaker
E.S. Ilhan, G.B. Akkus, A. B. Bener 698
- A development platform for distributed user interfaces
Anders Larsson, Magnus Ingmarsson, Bo Sun (S) 704

Security

- A Dynamical System Approach to Intrusion Detection Using System Call Analysis
Nitin Kanaskar, Remzi Seker, S. Ramaswamy 710
- Multi-level Anomaly Detection with Application-Level Data
Swapna S. Gokhale, Jijun Lu 718

Grid Technology

- A Four-layered Semantic Grid Architecture
Celia Ghedini Ralha, Jose Nelson C. Allemand, Alba C. M. Melo 724

Software Metrics, Measurement and Evaluation

Performance Analysis of the Active Object Pattern in Middleware <i>Paul J. Vandal, Swapna S. Gokhale, Aniruddha S. Gokhale</i>	730
Analyzing the Applicability of a Theoretical Model in the Evaluation of Functional Size Measurement Procedures <i>Nelly Condori-Fernandez, Oscar Pastor (S)</i>	736
Software Documents: Comparison and Measurement <i>Tom Arbuckle, Adam Balaban, Dennis K. Peters, Mark Lawford (S)</i>	740

Industrial Workshop

Workflow Management and Service Oriented Architecture <i>Theodorich Kopetzky, Dirk Draheim</i>	749
Implementing Agile Development - More than Changing Methodology <i>Chuck Fredrick</i>	751
Knowledge Modelling using UML <i>A. J. Rhem</i>	755

Reviewers' Index	758
-------------------------------	-----

Authors' Index	760
-----------------------------	-----

Note: (S) means short paper.

A Three Level Multi-agent Architecture to Foster Knowledge Exchange

Juan Pablo Soto, Aurora Vizcaíno, Javier Portillo-Rodríguez, Mario Piattini
*Alarcos Research Group, Information Systems and Technologies Department, UCLM-Soluziona
Research and Development Institute, University of Castilla – La Mancha
Ciudad Real, Spain*
*jpsoto@proyectos.inf-cr.uclm.es, {aurora.vizcaino, mario.piattini}@uclm.es,
javier.portillo@alu.uclm.es*

Abstract

This paper proposes a multi-agent architecture based on the concepts of communities of practice and reputation to manage knowledge management systems. The main goal of this proposal is to emulate the behavior of communities of practice where people exchange information and in this way attempt to foster the reuse of information in organizations which use knowledge base or knowledge management systems.

1. Introduction

The need to support knowledge processes in organizations has always existed. However, its importance has definitely increased in the last few years. Recently, the concept of knowledge management suggests a paradox since compared with traditional production factors knowledge is so complex, scattered and hidden that it is rather complicated to manage it.

On the other hand, traditional Knowledge Management Systems (KMS) have received certain criticism as they are often implanted in companies overloading employees with extra work; for instance, employees have to introduce information into the KMS and worry about updating this information. As a result of this, these systems are sometimes not greatly used by the employees since the knowledge that these systems have is often not valuable or on other occasions the knowledge sources do not provide the confidence necessary for employees to reuse the information. For this purpose, companies create both social and technical networks in order to stimulate knowledge exchange. An essential ingredient of knowledge sharing information in organizations is that of “community of practice”, by which we mean groups of people with a common interest where each member

contributes knowledge about a common domain [12]. The ability of a community of practice to create a friendly environment for individuals with similar interests and problems in which they can discuss a common subject matter encourages the transfer and creation of new knowledge. Many companies report that such communities help reduce problems caused by lack of communication, and save time by “working smarter”[13]. For these reasons, we consider the modelling of communities of practice into KMS as an adequate method by which to provide these systems with a certain degree of control to measure the confidence and quality of information provided by each member of the community.

In order to carry this out, we have designed a multi-agent architecture in which agents try to emulate human behaviour in communities of practice with the goal of fostering the use and exchange of information where intelligent agents suggest “trustworthy knowledge” to the employees and foster the knowledge flow between them.

The remainder of this work is organized as follows. The next section presents two important concepts that exist in the development of our work (agents and trust). In Section Three the multi-agent architecture proposed to manage trustworthy KMS is presented. In Section Four a prototype developed to evaluate our architecture is explained in order to illustrate how it could be used. Finally, conclusions are presented in Section Five.

2. Agents and trust

Because of the importance of knowledge management, tools to support some of the tasks related to knowledge management have been developed. Different techniques are used to implement these tools. One of them, which is proving to be quite useful, is

that of intelligent agents [10]. Software agent technology can monitor and coordinate events, meetings and disseminate information [1]. Furthermore, agents are proactive; this means they act automatically when it is necessary. The autonomous behavior of the agents is critical to the goal of this research since agents help to reduce the amount of work that employees have to perform. On the other hand one of the main advantages of the agent paradigm is that it constitutes a natural metaphor for systems with purposeful interacting agents, and this abstraction is close to the human way of thinking about their own activities [14]. This foundation has led to an increasing interest in social aspects such as motivation, leadership, culture or trust [3]. Our research is related to this last concept of "trust" since artificial agents can be made more robust, resilient and effective by providing them with trust reasoning capabilities.

For agents to function effectively in a community, they must ensure that their interactions with the other agents are trustworthy. For this reason it is important that each agent is able to identify trustworthy partners with which they should interact and untrustworthy correspondents with which they should avoid interaction. The stability of a community depends on the right balance of trust and distrust.

3. Our proposal

The goal of this work is to provide a reputation model for communities of practice using a multi-agent architecture that:

- Assists employees in identifying trustworthy entities.
- Gives artificial agents the ability to reason about the trustworthiness of other agents or of a knowledge source.
- Encourages knowledge exchange between the community members.
- Provides the confidence necessary to foster the usage of information and knowledge of the KMS.

To do this, we first need to define a conceptual model for the agent that permits it to obtain the level of confidence of an information source or of a provider of knowledge.

The conceptual model of the agent is based on two related concepts: trust and reputation. The former can be defined as confidence in the ability and intention of an information source to deliver correct information [2] and the latter as the amount of trust an agent has in an information source, created through interactions with information sources. There are other definitions for these concepts [4, 6]. However, we have presented

the most appropriate for our research since the level of confidence in a source is based on, in our case, previous experience of this.

The reputation of an information source not only serves as a means of belief revision in a situation of uncertainty, but also serves as a social law that obliges us to remain trustworthy to other people. Therefore, people, in real life in general and in companies in particular, prefer to exchange knowledge with "trustworthy people" by which we mean people they trust. People with a consistently low reputation will eventually be isolated from the community since others will rarely accept their justifications or arguments and will limit their interaction with them. It is for this reason that the remainder of this paper deals mainly with reputation.

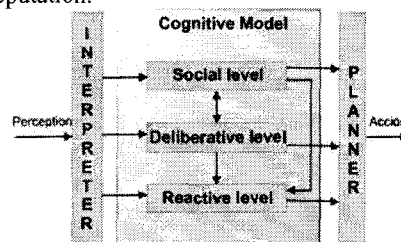


Figure 1. General architecture

Taking the concepts reputation and communities of practice into account we designed a multi-agent architecture which is composed of three levels (see Figure 1): reactive, deliberative and social. The reactive and deliberative levels are considered by other authors as typical levels that a multi-agent system must have [9]. On the other hand, the social level is not frequently considered in an explicit way, despite the fact that these systems (multi-agent systems) are composed of several individuals, interactions between them and plans constructed by them. The social level is only considered in those systems that try to simulate social behaviour or those that represent a more generic architecture prepared to represent this or other behaviour. Since we wish to emulate human feelings such as trust, reputation and even intuition we have added a social level that considers the social aspects of a community which takes into account the opinions and behaviour of each of the members of the community. Other previous works have also added a social level, for instance in [5] the author tries to emulate human emotions such as fear, thirst, bravery and also uses an architecture of three levels: reactive, deliberative and social.

In the following paragraphs we will explain each of these levels in detail.

Reactive level: This is the agent's capacity to perceive changes in its environment and to respond to

these changes at the precise moment at which they happen. It is in this level when an agent will execute the request of another agent without any type of reasoning. That is to say, the agent must act quickly in the face of critical situations.

Deliberative level: The agent may also have a behaviour which is oriented towards objectives, that is, it takes the initiative in order to plan its performance with the purpose of attaining its goals. In this level the agent would use the information that it receives from the environment, and from its beliefs and intuitions, to decide which is the best plan of action to follow in order to fulfill its objectives.

Social level: This level is very important as our agents are within communities and they exchange information with other agents. Thanks to this level they can cooperate with other agents by using an expressive language. This language analyzes the present situation, considering the goals and interests of the agent and structure solutions in the form of plans.

Two further important components of our architecture are the *Interpreter* and the *Planner*. The former is used to perceive the changes that take place. The planner indicates how the actions should be executed.

In this paper only the deliberative architecture is described due to space restrictions.

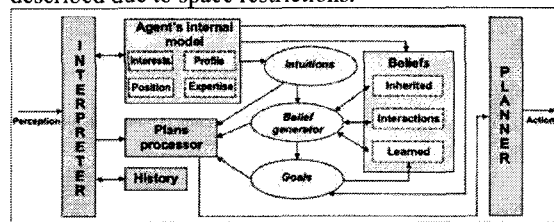


Figure 2. Deliberative architecture.

The components of the Deliberative Architecture are (see Figure 2);

Agent's internal model: As an agent represents a person in a community this model stores the user's features. Therefore, this module stores the following parts:

- The *interests*. This part is included in the internal model in order to make the process of distributing knowledge as fast as possible. That is, the agents are able to exchange knowledge automatically, checking whether their stored knowledge matches with the interests of other agents. This behaviour fosters knowledge sharing and reduces the amount of work employees have to do because they receive knowledge without making searches.
- *Expertise*. This term can be briefly defined as the skill or knowledge of a person who knows a great

deal about a specific thing. This is an important factor since people often trust in experts more than in novice employees.

- *Position*. Employees often consider information that comes from a boss as being more reliable than that which comes from another employee in the same (or a lower) position as him/her [11]. In an enterprise this position can be established in different ways, for instance by using an organizational diagram or classifying the employees according to the knowledge that a person has.

Such different positions inevitably influence the way in which knowledge is acquired, diffused and eventually transformed in the local area. Because of this these factor will be calculated in our research by taking into account a weight that can strengthen this factor to a greater or to a lesser degree.

History: This component stores the interactions of the agents with the environment.

Belief generation: This component is one of the most important of the cognitive model because it is in charge of creating and storing the agent's knowledge. Moreover, it defines the agent's beliefs.

Beliefs: The beliefs module is composed of the inherited beliefs of the organization, lessons learned, and agents' interactions. Inherited beliefs are the organization's beliefs that the agent receives. For instance: an organizational diagram of the enterprise, the expertise of each employee, the philosophy of the company or community. Lessons learned are the lessons that the agent obtains while it interacts with the environment This interaction can be used to establish parameters in order to know what the agent can trust (agents or knowledge sources).

Intuitions: The intuitions are beliefs that have not been verified but which it thinks may be true. According to [7] intuition has not yet been modelled by agent systems. In this work we have tried to adapt this concept by comparing the agents' profiles to obtain an initial value of intuition that can be used to form a belief about an agent.

Goals: The goals are formed by the objectives of the agent. For instance, one of the goals of each member of a community of practice is knowledge exchange. The goals are defined in accordance with the community or group in which the agent interacts

4. Prototype

In order to test our architecture we have developed a prototype system into which people can introduce documents and where these documents can also be

consulted by other people. The goal of this prototype is to allow software agents to help employees to discover the information that may be useful to them thus decreasing the overload of information that employees often have and strengthening the use of knowledge bases in enterprises. In addition, we try to avoid the situation of employees storing valueless information in the knowledge base.

The main feature of this system is that when a person searches for knowledge in a community, and after having used the knowledge obtained, that person then has to evaluate the knowledge in order to indicate whether:

- The knowledge was useful.
- How it was related to the topic of the search (for instance a lot, not too much, not at all).

To design this prototype we have designed a *User Agent* and a *Manager Agent*. The former is used to represent each person that may consult or introduce knowledge in a knowledge base. Therefore, the *User Agent* can assume three types of behavior or roles similar to the tasks that a person may carry out in a knowledge base. The User Agent plays one role or another depending upon whether the person that it represents carries out one of the following actions:

- The person contributes new knowledge to the communities in which s/he is registered. In this case the User Agent plays the role of **Provider**.
- The person uses knowledge previously stored in the community. Then, the User Agent will be considered as a **Customer**.
- The person helps other users to achieve their goals, for instance by giving an evaluation of certain knowledge. In this case the role is of a **Partner**. So, Figure 3 shows that in community 1 there are two User Agents playing the role of Partner, one User Agent playing the role of Consumer and another being a Provider.

The second type of agent within a community is called the *Manager Agent* (represented in black in Figure 3) which must manage and control its community.

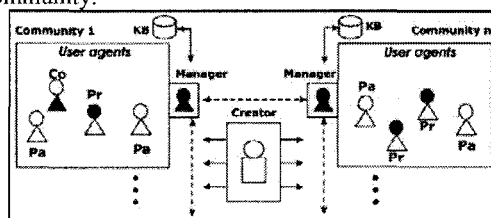


Figure 3. Communities of agents

The prototype provides the options of using community documents and updating reputation values,

proposing new topics in the community, etc. Due to space limitations, we shall now describe only the following situation:

Using community documents and updating reputation values. People can search for documents in every community in which they are registered. When a person searches for a document relating to a topic his/her User Agent consults the Manager Agent about which documents are related to their search. Then, the Manager Agent answers with a list of documents. The User Agent sorts this list according to the reputation value of the authors, which is to say that the contributions with the best reputations for this Agent are listed first. On the other hand, when the user does not know the contributor then the User Agent consults the Manager Agent about which members of the community know the contributors. Thus, the User Agent can consult the opinions that other agents have about these contributors, taking advantage of other agents' experience. To do this the Manager consults its interaction table and responds with a list of the members who know the User Agent. Then, this User Agent contacts each of them. If nobody knows the contributors then the information is listed, taking their authors' expertise and positions into account. In this way the User Agent can detect how worthy a document is, thus saving employees' time, since they do not need to review all the documents related to a topic but only those considered most relevant by the members of the community or by him/herself according to previous experience with the document or its authors.

Once the person has chosen a document, his/her User Agent adds this document to its own document list (list of consulted documents), and if the author of the document is not known by the person because it is the first time that s/he has worked with him/her, then the Community Manager adds this relation to the interaction table. This step is very important since when the person evaluates the document consulted, his/her User Agent will be able to assign a trustworthy value to that document. The formulas used to approach this have been explained in [8] (they have been omitted due to space constrains).

5. Conclusions

Communities of practice have the potential to improve organizational performance and facilitate community work. Because of this we consider it important to model people's behavior within communities with the purpose of imitating the exchange of information in companies that are

produced in those communities. Therefore, we are attempting to encourage the sharing of information in organizations by using knowledge bases. To do this we have designed a multi-agent three-layer architecture where the artificial agents use similar parameters to those of humans in order to evaluate knowledge and knowledge sources. These factors are: reputation, expertise, position, previous experience and even intuitions.

This approach implies several advantages for organizations as it permits them to identify the expertise of their employees and to measure the quality of their contributions. Therefore, it is expected that a greater flow of communication will exist between them which will consequently produce an increase in their knowledge.

6. Acknowledgement

This work is partially supported by the ENIGMAS (PIB-05-058), and MECENAS (PBI06-0024) project, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, both in Spain. It is also supported by the ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/ Fondos Europeos de Desarrollo Regional (FEDER) in Spain.

7. References

- [1] Balasubramanian, S., Brennan, R. and Norrie, D., An Architecture for Metamorphic Control of Holonic Manufacturing Systems. *Computers in Industry*, Vol. 46, No. 1, 2001, pp. 13-31.
- [2] Barber, K. and Kim, J., Belief Revision Process Based on Trust: Simulation Experiments. In *4th Workshop on Deception, Fraud and Trust in Agent Societies*, Montreal Canada, 2004.
- [3] Fuentes, R., Gómez-Sanz, J. and Pavón, J., A Social Framework for Multi-agent Systems Validation and Verification. Wang, S. et al Eds. *ER Workshops 2004*, Springer-Verlag, LNCS 3289, 2004, pp. 458-469.
- [4] Gambetta, D., Can We Trust Trust? In *Gambetta, D., ed.: Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, New York, 1990, pp. 213-237.
- [5] Imbert, R. and de Antonio, A., When Emotion Does not Mean Loss of Control. In *Lecture Notes in Computer Science*, T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, Eds. Springer-Verlag, London, 2005, pp. 152-165.
- [6] Marsh, S., Formalising Trust as a Computational Concept. PhD Thesis, University of Stirling, 1994.
- [7] Mui, L., Halberstadt, A. and Mohtashemi, M., Notions of Reputation in Multi-Agents Systems: A Review. *International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'02)*, 2002, pp. 280-287.
- [8] Soto, J.P., Vizcaino, A., Portillo, J. and Piattini, M., Knowledge Management Systems with Reputation and Intuition: What for?, *Accepted to be published in International Conference on Enterprise Information Systems (ICEIS'07)*, 2007.
- [9] Ushida, H., Hirayama, Y. and Nakajima, H., Emotion Model for Life like Agent and its Evaluation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI'98 / IAAI'98)*. 1998. Madison, Wisconsin, USA, 1998, pp. 62-69.
- [10] van-Elst, L., Dignum, V. and Abecker, A., Agent-Mediated Knowledge Management. In *International Symposium AMKM 2003*, Stanford, CA, USA, Springer, 2003.
- [11] Wasserman, S. and Glaskiewicz, J., *Advances in Social Networks Analysis*. Sage Publications, 1994.
- [12] Wenger, E., *Communities of Practice: Learning Meaning, and Identity*, Cambridge U.K.: Cambridge University Press, 1998.
- [13] Wenger, E., McDermott, R., and Snyder, W., *Cultivating Communities of Practice*, Boston: Harvard Business School Press, 2002.
- [14] Wooldridge, M. and Ciancarini, P., Agent-Oriented Software Engineering: The State of the Art. In *Wooldridge M., Ciancarini, P. (Eds.), Agent Oriented Software Engineering*, Springer-Verlag, LNAI 1975, 2001.