IJICT International Journal of Information & Communication Technology

ITRC

Volume 1- Number 1- March 2009 (pp. 49-57)

An Agent-Mediated Knowledge Management System in Call Centers Using SMV and TROPOS

Kaveh Pashaei
MAS Research Group,
Electrical and Computer Engineering Faculty
School of Engineering, University of Tehran
Tehran, Iran
K.Pashaei@ece.ut.ac.ir

Fattaneh Taghiyareh
MAS Research Group,
Electrical and Computer Engineering Faculty
School of Engineering, University of Tehran
Tehran, Iran
FTaghiyar@ut.ac.ir

Farzad Peyravi
MAS Research Group,
Electrical and Computer Engineering Faculty
School of Engineering, University of Tehran
Tehran, Iran
HF.Peyravi@ece.ut.ac.ir

Abstract— **Call centers have become one of the most cost effective ways of selling products to customers and giving services to them in different industries. By applying knowledge management solutions, we can meet call centers' challenges and gain benefits of reduced training costs, improved call handling and greater flexibility. This paper describes an agent mediated knowledge management system in call centers using the Tropos methodology. We use structure-in-5 for architectural design which specifies that our KM system is an aggregation of five sub-structures. Furthermore we develop a formal methodology and technique to verify the validity of communication protocols defined in a multi-agent environment. This is accomplished by examining agent conversations before deploying the system. The methodology leads to the definition of six different classes of agents. Our experiments develop proof of concept module for a call center that automatically verifies some of the important properties identified in this methodology. Results prove the agent's specification and indicate that our proposed model works accurately. The paper concludes with observations on the call centers and the role of agents in the proposed model.**

Keywords-component; Call Center; Tropos, Methodology; Knowledge Management; Multi-agent System; SMV;

I. INTRODUCTION

A call center is an organizational unit where inbound calls are received or outbound calls placed for the purposes of sales, support, advice and other business transactions [1]. In recent years, call centers have gained popularity as cost-effective avenues for selling and servicing customers and the primary way of interacting with them [2]. Although call centers provide many business advantages including: improved efficiency, reduced costs and greater flexibility, a call center is faced with consider-able challenges such as wide range of customers, expect instant answers to questions, complex knowledge to be learned and high staff turnover [3]. By applying knowledge management solution, we can meet these challenges and gain benefits of reduced training costs, improved call handling and greater flexibility.

In [3] benefits of knowledge management in call centers are declared. Two aspects to knowledge management are covered: the efficient processes that must be put in place and how to establish a knowledge management system to support these processes. An evaluation on need for knowledge management in a call center for improving quality of customer services, by addressing the issues specifically relating to information and knowledge management is discussed in [4]. Their model was an application of knowledge management in call centers. In [1] they suggested a knowledge-based hierarchy of 'advice-type' call centers and discussed associated knowledge management strategies for different sized centers in order to support complexity of knowledge management process caused by growing size.

In another work, the way front-line staffs such as call center staff operate, and how knowledge management can be used to meet their needs is analyzed [5]. They did not provide the final answers to the challenges of bringing improved knowledge management to front line staff.

Guizzardi and Perini discuss the use of the Agent Organization paradigm as basis for the development of a support system for Knowledge Management (KM). They present a strong claim to the importance of the initial phases of a system's development, aiming at grasping the requirements of the system to be, both in terms of the individual perspective of the organizational members and the overall objectives of the organization. This analysis process rests on an iterative workflow in which agent-oriented modeling plays a crucial role in understanding the domain's organization stakeholder's needs for KM systems; basically, by tracing system requirements back to the stakeholder's goals [18].

[19] Describes an agent-oriented methodology based on Tropos for the analysis and design of KM Systems that offers appropriate abstractions for modeling and designing the characteristics of the organizational setting of the system. The method is illustrated using a fictitious scenario where a newcomer in a knowledge organization decides to join an existing Community of Practice (CoP) in order to share knowledge and adjust to his new working environment.

Fuxman and etc [20], propose a new specification language, called Formal Tropos, that offers the primitive concepts of early requirements frameworks (actor, goal, strategic dependency), but supplements them with a rich temporal specification language. They also extend existing formal analysis techniques, and in particular model checking, to allow for an automatic verification of relevant properties of the early requirements specification. Their preliminary experiments show that formal analysis reveals gap sand inconsistencies in early requirements that are by no means trivial to discover without the help of formal analysis tool.

In previous works they described the role of knowledge management in call centers and the benefits that can be expected. All of above researches did not work on multi-agent system to handle the complexity and distribution of knowledge in call centers. A general model based on software agents design by an agent methodology to apply in call center is desirable. Pervious works consider only early requirement analysis phase and they don't prove their proposed model.

The distributed nature of knowledge in call centers and other department of organization and complexity of tacit knowledge lead us to use multi-agent systems to deal with theses features. Distribution of expertise, problem solving capabilities and responsibilities are consequences of division of labour in modern companies. Also both generation and use of knowledge are not evenly spread within the organization.

This paper describes a solution to agent mediated knowledge management system in call centers using Tropos methodology and then verifies it with NuSMV model checker. We presented all models required in this methodology for developing the multi-agent system. The outline of the paper is as follows. In section 2 a brief overview about call center is summarized and then we demonstrate our designed model. In section 3 we illustrate the Tropos methodology. NuSMV model checker is described in section 4. Our related models and diagrams are shown in section 5. In section 6 we verified some agent's properties. Finally we conclude in Section 7.

II. Basic Motive behind the Proposed Model for Call Centers

Call centers are the point of entry for most customer communication and have gained popularity as a link for selling and servicing customers. Organizations are now realizing the critical importance of every customer contact and are starting to understand the potential of the call center for becoming the focal point for customer relationship management strategies [2].

In their attempt to be truly effective in implementing CRM in the call centers, organizations often find themselves wrestling with a number of challenges [2]. The most urgent challenge facing call centers today is the fact that it is becoming increasingly difficult and costly to recruit and retain qualified call center agents, and the associated training costs can be overwhelming. In addition, there is a real risk that inexperienced agents may not be able to provide the level of service that customers deserve and demand. Identifying the right products and services for agents to sell to customers presents yet another challenge.

When call center agents responding to customer inquiry, or a customer using self-service, call center agents need efficient access to the right data, information or knowledge in order to obtain fast, accurate and consistent answers. Another challenge is the provision of the right knowledge for call center staff to handle inquiries in an adequate timeframe.

Our design is based on Nanoka's SECI model. Nonaka defines types of knowledge as tacit or explicit

[6]. The SECI cycles are based on the assumption that knowledge is created through conversion between tacit and explicit knowledge. Four tacit and explicit knowledge conversion mechanisms are: socialization, externalization, internalization, and combination [6]. Figure 1 shows an overall diagram that describing the status of our proposed model.

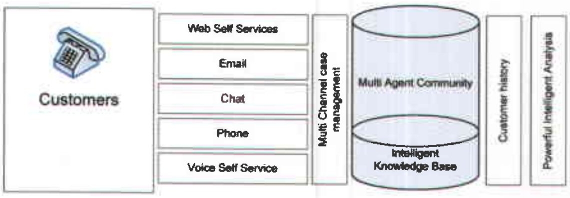

Figure 1. Model of multi-agent call center

By applying knowledge management solutions we can meet call centers challenges. In our model we designed a knowledge management (KM) system that exploit Nanoka model by applying multi-agent system technology. Organizations are decomposed into some organizational units and one of the most important of them is call center. Employees in units of organization have their expertise and knowledge. In our model each unit is a group that employees are its members. Each employee or manager belongs to only one group, a group header is an agent that controls the whole operations of group and knows all of members and their abilities and knowledge. Group header also knows the other groups and their knowledge. In our model each employee communicates with each other and sends, receives questions, offers, suggestions, documents and etc. When a call center agent or each employee facing a problem or a question and needs some knowledge, the system finds best person and redirect question to him.

We proposed a multi-agent system for socialization, between employees and call center agents, externalization, internalization and combination. Each employee working in a department can meet his/ her requiring knowledge by connecting the right people at right time. Employees can internalize explicit knowledge into tacit knowledge and externalize tacit knowledge to explicit knowledge. Moreover members of different groups can send suggestion, offer and related documents which may be useful for other members to share knowledge.

III. METHODOLOGY

A. *Requirement Analysis*

Requirement analysis represents the initial phase in most software engineering methodologies. Requirement analysis in Tropos consists of two phases: Early Requirements and Late Requirements analysis. Early requirement is concerned with understanding the organizational context within which the system-to-be will eventually function. Late requirement analysis, on the other hand, is concerned with definition of the functional and non-functional requirements of the system-to-be [7].

During the early requirement analysis, the requirement engineer identifies the domain stakeholders and models them as social actors who depend on one another for goal to be fulfilled, tasks to be performed, and resources to be furnished. Through these dependencies, one can answer why questions, in addition to what and who, regarding system functionality. Answers to why questions ultimately link system functionality to stakeholder needs, preferences, and objectives. Actor diagram and rationale diagram are used in this phase.

During the late requirement analysis, the conceptual model developed during early requirements is extended to include the system-to-be as a new actor, along with dependencies between these actors and others in its environment. These dependencies define functional and nonfunctional requirements for the system-to-be. Actor diagrams and rationale diagrams are also used in this phase.

B. *Architectural Design*

System architectural design has been the focus of considerable research during the last 15 years that has produced well-established architectural styles and frameworks for evaluating their effectiveness with respect to particular software qualities. Examples of styles are pipes-and-filters, event-based, layered, control loops, and the like [8]. In Tropos, we are interested in developing a suitable set of architectural style for multi-agent system (MAS) are intentional and social, rather than implementation-oriented, we turn to theories that study social structures that result from a design process, namely, organization Theory and Strategic Alliances. Organizational Theory describes the structure and design of an organization; Strategic Alliances model the strategic collaboration of independent organizational stakeholders who have agreed to pursue a set of business goals [7].

Structure-in-5 [9] specifies that an organization is an aggregate of five sub structures. At the base level sits the Operational Core, which carries out the basic tasks and procedures directly linked to the production of products and services (acquisition of inputs, transformation of inputs into outputs, distribution of outputs). At the top level lies the Strategic Apex, which makes executive decisions ensuring that the organization fulfills its mission in an effective way and defines the overall strategy of organization in its environment. The middle Line establishes a hierarchy of authority between the Strategic Apex and the Operational Core. It consists of managers responsible for supervising and coordinating the activities of the Operational Core. The Technostructure and the Support are separated from the main line of authority and the influence the operational core only indirectly. The Technostructure serves the organization by making the work of others more effective, typically by standardizing work process, outputs, and skills. It is also in charge of applying analytical procedures to adapt the organization to its operational environment. The Support provides specialized services, at various levels of hierarchy, outside the basic operating work flow (e.g., legal counsel, R&D, payroll, cafeteria) [10].

C. *Detail Design*

System The detailed design phase is intended to introduce additional detail for each architectural component of a system. It consists of defining how the goals assigned to each actor are fulfilled by agents with respect to social patterns.

Social patterns in Tropos [8] are design patterns focusing on social and intentional aspects that recurrent in multi-agent and cooperative systems. In particular, the structures are inspired by the federated patterns introduced in [9] and [11]. These social patterns are classified into two categories: Pair and Mediation.

The pair patterns - such as booking, call-for-proposal, subscription, or bidding -describe direct interaction between negotiation agents. For instance, the bidding pattern involves an initiator and a number of participants. The initiator organizes and leads the bidding process, publishes the bid to the participants, and receives various proposals. In each iteration, the initiator can accept an offer, raise the bid, or cancel the process.

The Mediation patterns - such as monitor, broker, matchmaker, mediator, embassy, or wrapper – feature intermediary agents that help other agents to reach an agreement on an exchange of service. For instance, in the broker agent is an arbiter and intermediary that request services from provider to satisfy the request of consumer.

IV. NuSMV Model Checker

NuSMV [17] is a symbolic model checker which verifies the correctness of properties for a finite state system. The system should be modeled in the input language of NuSMV, called SMV, and the properties should be specified in CTL or LTL. The only data types in the language are finite ones, including Booleans, scalars and fixed arrays. A SMV code is a set of Module definitions, including a main module. Processes are instantiated from Modules, and are used to model interleaving concurrency. The program executes a step by non-deterministically choosing a process, then executing all of the assignment statements in that process in parallel. The main control structure in SMV is the next-case statement. Using this statement, the programmer can specify the next value of a variable, according to the current value of all variables in the code.

The main features of NUSMV are the following:

Functionalities: NUSMV allows for the representation of synchronous and asynchronous finite state systems, and for the analysis of specifications expressed in Computation Tree Logic (CTL) and Linear Temporal Logic (LTL), using BDD-based and SAT-based model checking techniques. Heuristics are available for achieving efficiency and partially controlling the state explosion. The interaction with the user can be carried on with a textual interface, as well as in batch mode.

Architecture: software architecture has been defined. The different components and functionalities of NUSMV have been isolated and separated in modules. Interfaces between modules have been provided. This reduces the effort needed to modify and extend NUSMV.

Quality of the implementation: NUSMV is written in ANSI C, is POSIX compliant, and has been debugged with Purify in order to detect memory leaks. Furthermore, the system code is thoroughly commented. NUSMV uses the state of the art BDD package developed at Colorado University, and provides a general interface for linking with state-of the-art SAT solvers. This makes NUSMV very robust, portable, efficient, and easy to understand by other people than the developers.

V. The Modeled System

In this section we will describe our model. We will mention only one example for each model.

A. Early Requirement Analysis

The first phase is the early requirements analysis. Figure 2 depicts the actor diagram of KM System. The main actors are Customers, Call Center Agent and Expert Agent. Customers depend on Call Center Agent to fulfill his/her goal: Answer Questions. Conversely, Call Center Agent depends on Customer to increase Customer's Satisfaction. Since the dependum satisfaction can not be defined precisely, it is represented as a soft goal. Furthermore, Call Center Agent depends on Expert Agent to provide Best Answer in a continuous way and get Answer (resource dependency). Also Call Center Agent depends on Expert Agent to Manage Profiles and Suggest to Employee.

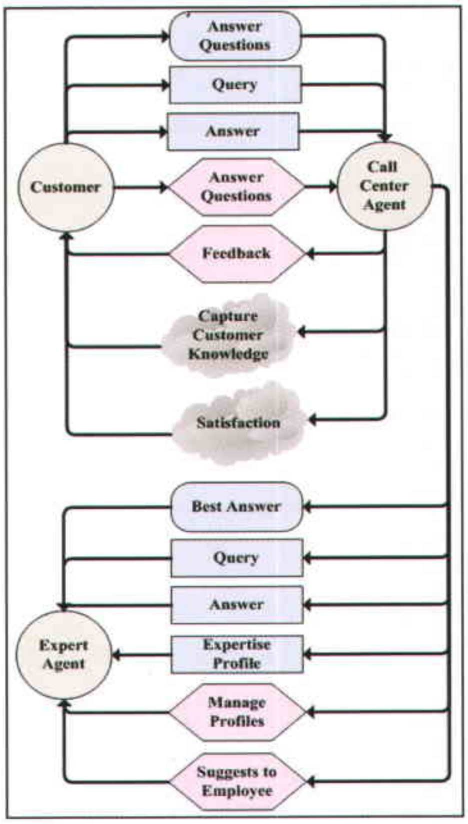

Figure 2. Actor diagram for KM System

Figure 3 focuses on one of the actor namely, Call Center Agent. To investigate that actor, the analysis proposes a goal Find Best Person that can be fulfilled by means of three sub-tasks. Tasks are partially ordered sequences of steps intended to accomplish some goals. In Tropos, tasks can be decomposed into sub-tasks and also goals, whose collective fulfillment completes the tasks. In Figure 3, Find Best Person is decomposed into tasks Search in Profile, Analyze the Question and Find Field of Question, which together accomplish the top-level task. Subgoals and subtasks can be specified more precisely through refinement. These decompositions eventually allow us to identify actors who can accomplish a goal, carry out a task, or deliver some needed resource for KM system.

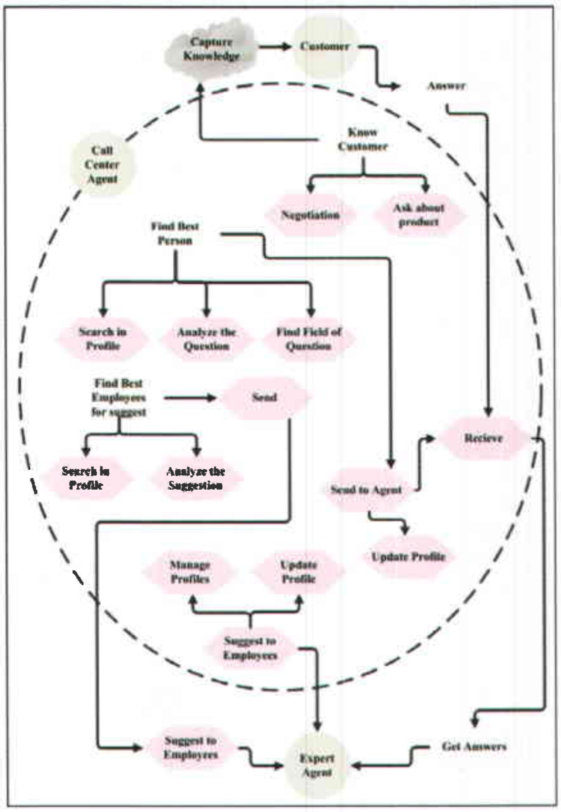

Figure 3. Means-Ends Analysis

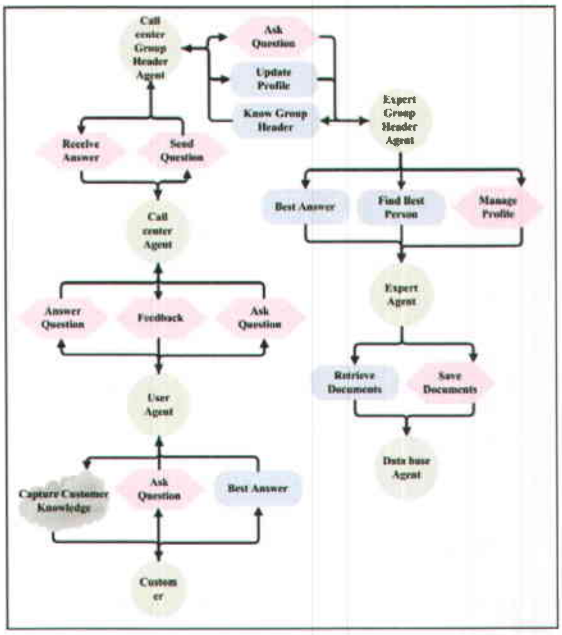

Figure 4. Refined Actor Diagram for the KM System

B. Late Requirement Analysis

In this section we describe late requirement analysis phase. In this analysis, the KM system is viewed as a full-fledged actor in the actor diagram depicted in figure 4.

With respect to the actors previously identified, Customer Agent depends on User Agent to Find Best Answer, while User Agent depends on Customer Agent to Capture Customer Knowledge. In this phase we introduce new actors, namely, User Agent, Call Center Group Header Agent, Expert Group Header Agent and Data Base Agent. Call Center Group Header Agent depends on Expert Group Header Agent for Ask Question, conversely Expert Group Header Agent depends on Call Center Group Header Agent to achieve Know Group Header goal. Furthermore Expert Agent depends on Data Base Agent to fulfill Save Document task and Retrieve Document goal. Although an actor diagram provides hints about why processes are structured in a certain way, it does not sufficiently support the process of suggesting, exploring, and evaluating alternative solutions. As late requirement analysis proceeds, KM system is given additional responsibilities an ends up as the dependee of several dependencies.

C. Architectural Design

Figure 5 suggest the possible assignment of system responsibilities for KM system following the structure-in-5 style. It is decomposed into six principal actors User Agent, Data Base Agent, Expert Agent, Expert Group Header Agent, Call Center Group Header Agent and Data Base Agent.

User Agent, Call Center Agent and Expert Agent Serve as the Operational Core. Expert Agent interacts primarily with Data Base Agent for Retrieve and Save Documents. User Agent negotiates with Call Center Agent for accomplishment of Ask Question and Retrieve Answer tasks. Data Base Agent constitutes the Support component. It provides Documents for Expert Agent. At the Middle Line, the Call Center Group Header Agent and Expert Group Header Agent assume the central position of the architecture. Expert Group Header Agent depends on Expert Agent to Find Best Person and Manage Profiles.

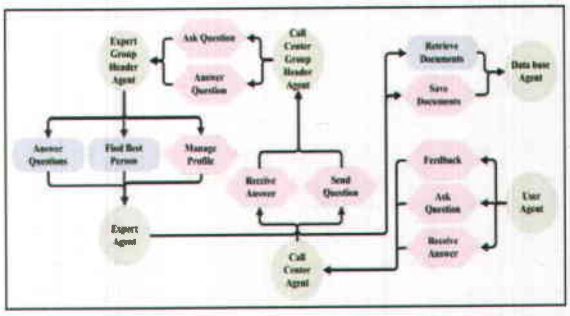

Figure 5. The KM System Architectural Design

D. Detailed Design

Finally, as mentioned before, the last model in the Tropos methodology is the Detailed Design. Figure 6 shows a Detailed Design diagram developed for the Expert Group Header Agent. The Expert Group

Header Agent and the dependencies are decomposed into a combination of social patterns, involving agents, pattern agents, subgoals, and subtasks. The Profile Manager deals with Register Agents and Manage Profile subtasks, on the other hand Communicator provides Feedback to Profile Manager social pattern. The Communicator depends on Finder to perform Analyze Question task and Find Best Person subgoal.

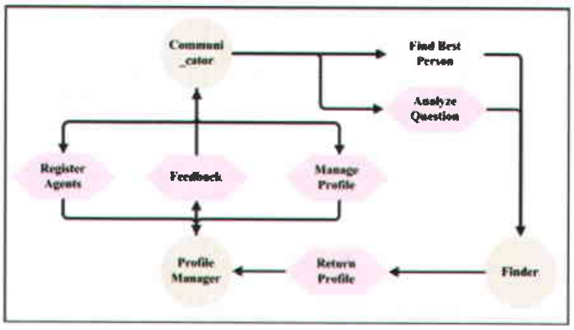

Figure 6. Detailed Design for the Expert Group Header Agent

VI. Model Checking

In order to verify our system, we model it by SMV and verify by its model checker. Considering features such as agent based and distribution of users and agents in multi-agent systems allow us to use SMV to model it. For automatic verification of relationships between dynamic properties of components of different aggregation levels by means of model checking techniques, a corresponding to the behavioral specification of the lower aggregation level representation of a finite state transition system should be translated into the input format of one of the existing model checkers. The model checker SMV has been chosen as a verification tool for two reasons. First, the input language of SMV is syntactically and semantically similar to the general description of a finite state transition system, which facilitates automatic translation into the SMV input format. Second, SMV uses efficient symbolic algorithms to traverse a model and the expressive temporal logic CTL for specifying properties to check.

In this section we describe our model and its protocols. Our model for co-operative information gathering is considered at two aggregation levels. At the higher level the multi-agent system as a whole is considered. At the lower level the four components and their interactions are considered: two information gathering agents A and B, agent C, and environment component E representing the conceptualized part of the external world. Each of the agents is able to acquire partial information from an external source (component E) by initiated observations. Each agent can be reactive or proactive with respect to the information acquisition process. An agent is proactive if it is able to start information acquisition independently of requests of any other agents, and an agent is reactive if it requires a request from some other agent to perform information acquisition.

Observations of any agent taken separately are insufficient to draw conclusions of a desired type, but the combined information of both agents is sufficient. Therefore, the agents need to co-operate to be able to draw conclusions. Each agent can be proactive with respect to the conclusion generation, i.e., after receiving both observation results an agent is capable to generate and communicate a conclusion to agent C. Moreover, an agent can be request pro-active to ask information from another agent, and an agent can be pro-active or reactive in provision of (already acquired) information to the other agent.

To analyze the model we use NUSMV verifier and for verify it we specified some goals. Generally these goals are extracted from requirement of our system. These goals are expressed in property format to meet SMV requirements.

The first property which is consider for our model is Effectiveness, which means that, each user in call center system have the ability to send and receive questions from all agents that exist in system. This property is expressed in CTL logic.

Effectiveness_1 Property

SPEC

AG(A_output_request_from_to_for_A_B_info -> AF(B_input_request_from_to_for_A_B_info))

Effectiveness_2 Property

SPEC

EF(A_output_communicated_send_from_to_A_C_info)

the results of this property involves iterations, user time, system time, BDD nodes allocated and model checking results is summarized in Table 1 and Table 2.

TABLE I. Model Checking Result by SMV for Effectiveness_1 Property

(AF B_input_request_from_to_for_A_B_info)	
iteration 0	3
iteration 1	21
iteration 2	19
iteration 3	15
iteration 4	4
iteration 5	1
user time	0.234375 s
system time	0.015625 s
BDD nodes allocated	60873
Model checking Result	true

TABLE II. Model Checking Result by SMV for Effectiveness_2 Property

(EF A_output_communicated_send_from_to_A_C_info)	
iteration 0	3
iteration 1	12
iteration 2	11
iteration 3	10
iteration 4	10
iteration 5	16
iteration 6	17
iteration 7	14
iteration 8	15
iteration 9	15
iteration 10	17
iteration 11	16
iteration 12	18
iteration 13	22
iteration 14	26
iteration 15	28
iteration 16	28
iteration 17	27
iteration 18	26

iteration 19	22
iteration 20	18
iteration 21	14
iteration 22	1
user time	0.609375 s
system time	0.03125 s
BDD nodes allocated	60873
Model checking time	0.593750 s
Model checking Result	true

The second property that is checked for the modeled system is reactiveness. This property says that each agent responsible for received questions and should send the answer to destination agent. This answer could be achieved via neighbor agent, database or by himself. This property is expressed in CTL logic.

Reactiveness_1 Property

SPEC

AG(A_output_communicated_send_from_to_A_C_info -> AF(C_input_communicated_send_from_to_A_C_info))

Reactiveness_2 Property

SPEC

AG(B_input_request_from_to_for_A_B_info -> EF(E_output_observed_provide_result_from_to_E_B_info))

The results of this property involves iterations, user time, system time, BDD nodes allocated and model checking results is summarized in Table 3 and Table 4.

TABLE III. MODEL CHECKING RESULT BY SMV FOR REACTIVENESS_1 PROPERTY

(AF C_input_communicated_send_from_to_A_C_info)	
iteration 0	3
iteration 1	8
iteration 2	7
iteration 3	6
iteration 4	3
iteration 5	12
iteration 6	11
iteration 7	10
iteration 8	10
iteration 9	16
iteration 10	17
iteration 11	14
iteration 12	15
iteration 13	19
iteration 14	21
iteration 15	20
iteration 16	22
iteration 17	26
iteration 18	30
iteration 19	37
iteration 20	37
iteration 21	38
iteration 22	37
Iteration 23	33
Iteration 24	29
Iteration 25	29
user time	0.296875 s
system time	0.015625 s
BDD nodes allocated	60873
Model checking time	0.293750 s
Model checking Result	true

TABLE IV. MODEL CHECKING RESULT BY SMV FOR REACTIVENESS_2 PROPERTY

(EF E_output_observed_provide_result_from_to_E_B_info)	
iteration 0	3
iteration 1	1

user time	0.546875 s
system time	0.015625 s
BDD nodes allocated	60873
Model checking time	0.55875 s
Model checking Result	true

The last property that is checked for the modeled system is proactiveness. This property states that agent should be autonomous. They must decide to send and receive message without any request from other agents. Agents should not simply act in response to their environment; they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate. This property is expressed in CTL logic.

Proactiveness_1 Property

SPEC

AG (E_output_observed_provide_result_from_to_E_A_info & E_output_observed_provide_result_from_to_E_B_info -> AF (C_input_communicated_send_from_to_A_C_info))

Proactiveness_2 Property

SPEC

AG(EF(A_input_provided_result_from_to_E_A_info & B_input_request_from_to_for_A_B_info) -> AF(A_output_communicated_send_from_to_A_C_info))

The results of this property involves iterations, user time, system time, BDD nodes allocated and model checking results is summarized in Table 5 and Table 6.

TABLE V. MODEL CHECKING RESULT BY SMV FOR PROACTIVENESS_1 PROPERTY

(AF C_input_communicated_send_from_to_A_C_info)	
iteration 0	8
iteration 1	7
iteration 2	6
iteration 3	3
iteration 4	12
iteration 5	11
iteration 6	10
iteration 7	10
iteration 8	16
iteration 9	17
iteration 10	14
iteration 11	15
iteration 12	19
iteration 13	21
iteration 14	20
iteration 15	22
iteration 16	26
iteration 17	30
iteration 18	37
iteration 19	37
iteration 20	38
iteration 21	37
iteration 22	33
iteration 23	29
iteration 24	29
iteration 25	28
user time	0.234375 s
system time	0.015625 s
BDD nodes allocated	60873
Model checking time	0.243750 s
Model checking Result	true

TABLE VI. MODEL CHECKING RESULT BY SMV FOR PROACTIVENESS_2 PROPERTY

(EF

(A_input_provided_result_from_to_E_A_info&B_input_request_from_to_for_A_B_in	
iteration 0	4
iteration 1	41
iteration 2	45
iteration 3	43
iteration 4	43
iteration 5	54
iteration 6	42
iteration 7	32
iteration 8	24
iteration 9	1
user time	0.3125 s
system time	0.015625 s
BDD nodes allocated	60873
Model checking time	0.443750 s
Model checking Result	true

VII. CONCLUSION

In this paper we presented one aspect of our research aimed at applying knowledge management to overcome call center challenges caused by different customer expectations such as instant answers to questions, complex knowledge to be learned as well as higher staff turnover. We proposed a model based on Nanoka SECI model by employing multi-agent systems. Each organizational unit creates a group of employees and each employee working in a department can meet his/ her requiring knowledge by connecting the right people at the right time. Our designed system can find the best employee with the highest expertise to answer the question.

We have analyzed this model using Tropos methodology. We believe that the methodology is particularly appropriate for generic and knowledge management systems including our application that can be used in a variety of operating environment and computing platform. The requirement-driven approach, on which Tropos is based, suggests that the methodology complements well to proposals for agent-oriented programming environments. Our purpose is to promote an efficient utilization of social patterns in order to achieve a successful detailed and architectural design of KM multi-agent systems. The four tacit and explicit knowledge conversion mechanisms are: socialization, externalization, internalization, and combination which can be reached in our model.

Furthermore, we modeled our system using SMV and verified it by NuSMV to show that by using our system, knowledge sharing can be satisfied. We proved that in our system each user can access distributed knowledge among users in call center system. Our finding leads us to believe that agent's design objectives, namely effectiveness, reactiveness and proactiveness can thoroughly be covered by our proposed model. A major aspect that emerged from our analysis is that agent's properties could be satisfied through BDD node allocated and these properties are consistent with agent's plan.

Future research directions can be intended to evaluate the business practices within organizations to better recognize which user model variables cooperates major roles in their learning behaviors. Our developed system will be organized within these organizations to evaluate the system's ability to facilitate the learning process, in which trainees join workgroups, and in the future, other key organizational learning practices.

REFERENCES

[1] G. Timbrell, S. Koller and S. Lindstaedt, "A Knowledge Infrastructure Hierarchy Model for Call Centre Processes ", presented at Proceedings I-KNOW '04: 4th International Conference on Knowledge Management, Graz, Austria, 2004.

[2] "Call Center's Role in CIM", in Cincom Encompass, available at www.Cicnom.com.

[3] J. Robertson, "Knowledge management for call centers ", technical report, 2002, available at http://www.kmcenter.org/ArticleShow.asp?ArticleID=2377.

[4] S. C. L. Koh, A. Gunasekaran, A. Thomas and S. Arunachalam, "The application of knowledge management in call centres", journal of knowledge management, vol. 9, 2005.

[5] J. Robertson, "Knowledge management for front-line staff", technical report, 2003, available at http://www.steptwo.com.au/papers/kmc_callcentre.

[6] I. Nanoka, "The Knowledge Creation Company", Harward Business Review, Vol. 69, 1991.

[7] F. Giunchiglia, J. Mylopoulos and A. Perini, "The Tropos Software Development Methodology: Processes, Models and Diagrams," presented at Proceedings of the first international joint conference on autonomous agents and multi agent systems, Italy, 2002.

[8] T. T. Do, M. Kolp and A. Pirotte, "Social patterns for designing multi-agent systems", presented at proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering, SEKE'03, San Francisco, 2003.

[9] S. Hayden, C. Carrik and Q. Yang, "Architectural design patterns for multi agent coordination", presented at Proceedings of the 3rd International Conference on Autonomous Agents, Agents'99, Seattle 1999.

[10] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos and A.Perini, "TROPOS: An Agent-Oriented Software Development Methodology", Journal of Autonomous Agent and MultiAgent Systems, vol. 8 (3), pp. 203-236, 2004.

[11] M. Kolp, P. Giorgini, and J. Mylopoulos, "A goal-based organizational perspective on multi-agent architectures", presented at Proceedings of the 8th International Workshop on Intelligent Agents: Agent Theories, Architectures and Languages, ATAL'01, Seattle, 2001.

[12] H. Mitzberg, "Structure in fives: Designing effective organizations", Upper Saddle River, NJ: Prentice Hall, 1992.

[13] B. Henderson-Sellers and P. Giorgini, "Agent-Oriented Methodologies", US: Idea Group Publishing, 2005.

[14] L. V. Elst, and A. Abcker, "Towards Agent-Mediated Knowledge Management", presented at Proceedings of International Symposium AMKM Stanford, CA, USA., 2003.

[15] M. Shaw and D. Garlan, "Software architecture: Perspective on an emerging discipline", Upper Saddle River, NJ: Prentice Hall, 1996.

[16] B. Bergeron, "Essential of Knowledge Management", John Wiley & Sons, 2003.

[17] NuSMV user manual, available through http://nusmv.irst.itc.it/NuSMV/ userman/index-v2.html.

[18] R. S. S. Guizzardi, and A. Perini, "Analyzing Requirements of Knowledge Management Systems with the Support of Agent Organizations", In Journal of the Brazilian Computer Society (JCBS). Special Issue on Agents Organizations, vol. 11, n. 1, July 2005, page. 51-62.

[19] R. S. S.Guizzardi, A. Perini, V. Dignum, "Providing Knowledge Management Support to Communities of Practice through Agent-oriented Analysis", in Proceedings of the 4th International Conference on Knowledge Management (I-KNOW'04), Granz, Austria, (2004), available at http://wwwhome.cs.utwente.nl/~souza/publications/guizzardi-perinidignumiknow04.pdf, last visited 23-06-2004.

[20] A. Fuxman, M. Pistore, J. Mylopoulos, and P. Traverso, "Model checking early requirements specifications in Tropos", In International Symposium on Requirements Engineering, pages 174–181, Toronto, CA, 2001, IEEE Computer Society.

Kaveh Pashaei is currently PhD. Student in software engineering in Electrical and Computer Engineering faculty of University of Tehran. He is a member of Multi Agent research group in University of Tehran. His research interests include ERP and KM with multi agent approach, stochastic process, data mining, and pattern recognition and detection.

Fattaneh Taghiyareh graduated from Tokyo Institute of Technology with a PhD. degree in computer engineering in 2000. She is currently the head of Information Technology engineering group, Electrical and Computer Engineering faculty of University of Tehran. She has established many research groups such as Multi Agent research group (2004) and Information Technology foundations (2006). Her research interests include ERP and CRM with multi agent approach, personalized recommender systems, adaptive e-learning systems, and semantic web services.

Farzad Peyravi graduated from University of Tehran a M.S. degree in Software engineering in 2006. His research interests include KM with multi agent approach, database, data mining, and Information retrieval.