IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 35-43
www.iosrjournals.org

Agent Based Distributed Tutoring System modeling using Extended Tropos approach

Sampathkumar TT, Venkateswaran V, PeddiReddy Vishnu
sampathchari27@gmail.com, venkateswaran.veeran@gmail.com, swaroop214@gmail.com

Abstract: *Due to the availability of enormous amount of resources for the tutoring system, there is a need for the distribution of the resources when developing such tutoring system.There are several users inthe distributed system where there is aconcurrent access of the resources. In such scenario there should be a definite access control policies for the resources. Since the agents are suitable for the distributed system and these access control policies would restrict the agents in accessing the resources. In our work we have extended Tropos which is an agent oriented software development methodology.Our extension would classify the resources based on the types of resources by which it can be accessed by other agents. This extension would also consider the temporal constraints in accessingthe resources. The resources have been classified into four types and the temporal constraints have been classified into three types. Our approach provides a systematic way of development of the distributed systems than the formal UML modeling technique by capturing the security constraints and providing the access control policies in the modeling stage itself. In our work, the Agent based distributed Tutoring System(ABDTS) has been taken as a case study and we have modeled the system using the extended Tropos methodology and the results are evaluated. We also we provide a comparativeanalysis between the models developed using Tropos, Unified Modeling Language (UML) which could incur that the Tropos could be used where the agents goals and tasks are to be modeled explicitly.*
Keywords: *Requirement engineering, Agent based systems, Tropos, Adaptive software environment, Security Requirements*

I. Introduction

In day-to-day life there has been drastic development of the technology and so the software development process has also been moving towards the Agent oriented software engineering because of the need for open, heterogeneous, distributed environments. Agent Oriented Software Engineering (AOSE) has been proposed and applied in the past few years [1, 2]. Such modeling mostly starts with the requirement elicitation and ends with the detailed design. The dynamic requirement elicitation is not formally supported by UML; hence Troposis introduced to address the dynamic requirement modeling with the Phases: early requirement analysis, late requirement analysis, architectural design, and detailed design. Such modeling drastically provides a proper guideline for the development of the Agent oriented systems.

One of the major problems faced by agent-oriented software engineering is the failure to achieve a strong industry acceptance. The main reason hindering this acceptance is lack of an accepted process-oriented methodology for developing agent-based systems. An interesting solution to this problem is the use of approaches that allow us to customize processes based on different types of applications and development environments. One technique that provides such a rational approach for the construction of tailored methods is method engineering [6].

Tropos [3] [9] [12] [13] [14] [15] has been used for modeling the requirements of the agents by the goals and tasks in which the system's requirements are adaptive in nature. The specialty of Tropos is that it follows a requirement driven approach for software development, by identifying the stake holders, in early requirements, analyzing the goals, roles, tasks before modeling the system-to-be, which formally leads to the satisfaction of the stakeholder's goals and also provides the stability in the development phases and reduces the conflicts which usually arise between the designing phase and the coding phase.

The main motto of ABDTSis to provide personalized tutoring to the learners. The Tutoring system has to handle huge amount of data in order to provide adaptive contents. This includes audio, video, open source tools, text and internet. And also there is a need to store and access such distributed contents with minimum system performance and network usage. Most of the Agent based Tutoring systems [7] [8] [9] [29] [30] find difficulty of handling and managing contents that leads the system to an undesired state. Hence in this work the Agent based Distributed Tutoring system is considered which emphasize on distribution of resources and accessing the resources in a secured manner.

In this work we modeled Agent Based Distributed Tutoring System (ABDTS)which emphasis moreon securitybut Troposis not extended in thataspect, hence we extend the Tropos that provides a modeling for security of the resources and the temporal constraints of the agents. ABDTShas been modeled using extended Tropos

Rest of the paper is structured as follows. Section 2 describes the tropos and its extension.Section 3 deals with the development activities in Troposand theproposed extension of Tropos. Section 4illustrates the development of Agent BasedDistributed Tutoring System with appropriate diagrams by applying extended Tropos. Section 5 discusses theroles ofagents in our proposed ABDTS.Section 6 depicts the comparative analysis of models using Tropos (Extended)and UML. Finally the Conclusion is discussed in Section 7.

II. Motivation

In the recent days, due to the availability of the open source materials from various sources such as youtube [33], IIT and NPTEL [31], MIT open course ware [32] for developing the tutoring system in the institution. Hence there are plenty of resources available for the development of the tutoring system which paves the way for distributing the resources.In such distributed environment, the resources can be replicated to several copies and this replication policy is contained in a file which is stored in the master node. Since modifying the replication policy would lead the system to inconsistent state. Suppose if the system is determined to have a maximum of three replication copies in the distributed system, altering the policy would collapse the system even if the value is altered to five because of the less availability of the storage space in the system. Hence the replication policy files should be given a restrictive access in which only the specific agents can alter the value. There are two types of agents: local and global agents. There would be restriction for the local and global agents which is defined by the system. Hence we have classified the resources in to four types of resources which are described in the section 3.

In the ABDTS, the resources can be relocated to other nodes if there is less availability of space in the specified node. Hence while accessing the course materials the agent has to relocate to the specific node and copy the resources to the central node. In this scenario, many failures can occur such as network failure etc and the resources cannot be accessed. To overcome this situation we have defined several temporal constraints explained in section 3. The agents are assigned with the specific duration for several actions. These durations are common for all the agents in the distributed system. This includes replication time, lookup time, retrieval time etc which is elaborated in section 3.

Since Tropos has the capability of modeling the agent oriented system modeling from the early requirements to detailed design.Hence we have extended the Tropos methodology considering the security constraints of the resources and temporal constraints. Tropos has general representation for the resources but we have added the additional representation of the resources which would consider the security constraints on the resources.

III. Tropos and extended Tropos methodology

Using the extended Tropos methodology (proposed approach) for modeling ABDTS, we ended up with the following modules: Student module, Tutor module, Exam module, Adaptive storage module, Resource retrieval module, replication module. The functions of the modules depend on the policies and directories present in the distributed system. The Naming directory decides the naming of the resources stored in the distributed system, the access control policies determines level of access to the resources, the location directory maps the resources to its location even though the resources are replicated and the replication policy determines the number of copies to be replicated in the distributed system. In such a scenario it is mandatory to provide a secured way of modifying such directories and there is a need of modeling such security aspects in the distributed system.

1.1 Tropos Methodology

The phases of Tropos are described in Figure 1 which starts from the early requirements and ends with the implementation.

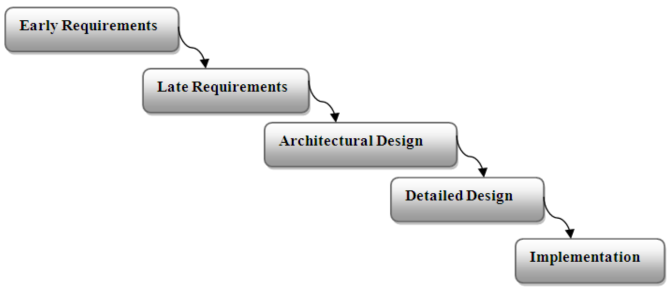

Figure1: Tropos Development Phase

Early Requirements: during this phase the relevant stakeholders in the system are identified, along with their respective roles
Late Requirements: the system-to-be is introduced as another actor and is related to stakeholder actors in terms of actor dependencies
Architectural Design: system actors are defined with further detail.
Detailed Design: In this phase the attributes of the agents and goals are modeled in detail.
Implementation: During this phase, the skeleton for the implementation phase is produced.

Troposconceptual models and diagrams are developed as instances of the following intentional and social concepts: actor, goal, dependencies, plan, resource, capability, and belief. Each of themis discussed below.

Actor: The notion of actor models an entity that has strategic goals and intentionality. An actor represents an agent (physical or software) as well as a role or a position
Goal: A goal represents the strategic interests of actors
Dependency: A dependency between two actors indicates that one actor depends on another in order to attain some goal, execute some plan, or deliver a resource
Plan: A plan represents a way of satisfying a goal
Resource: A resource represents a physical or informational entity that one actor wants and another can deliver
Capability: A capability represents the ability of an actor to define, choose and execute a plan to fulfill a goal
Belief: Beliefs are used to represent each actor's knowledge about the world
De-composition: And/OR decomposition of the goals/Tasks. And decomposition represents that all the sub-goals should be satisfied to attain the main goal

In Tropos, the temporal constraints of the goals/tasks on the resources are not explicitly modeled and the security constraints of the resources are not discussed. Hence we extend the Tropos methodology and apply it to the requirements of the distributed tutoring system.

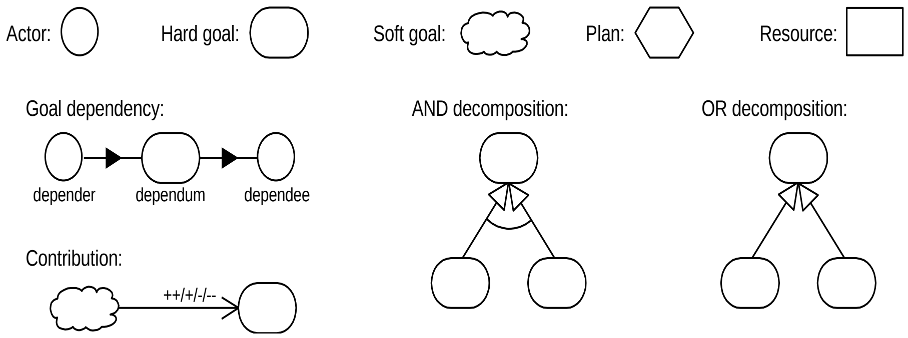

Figure 2. Representations in Tropos

1.2 Tropos extensions

Tropos has been extended for the requirement analysis of the business process in [12] [15]. The legal requirements of the organization and the software development with the legal constraints were discussed in [14]. But there is a lack of modeling of security requirements; hence we extend Troposconsidering the resource securities and access policies of the resources in the distributed environment in which we consider the temporal properties of the goals and tasks which are defined in [12].

In [18], the authors have described some vulnerability caused by the Multi-Agent System's (MAS) characteristics, such as cooperation, autonomy, and mobility. To address such vulnerabilities, they extend the original FAME4 Agent-oriented Modeling Language (FAML) by inserting security techniques such as an interaction history log [18].

Various access control policies, authentication and authorization policiesare described in [17]. The traditional access control policy such as Discretionary Access Control(DAC) and Mandatory Access Control (MAC) policies define the access control in the system. In DAC, authorization is controlled at the discretion of users, who are the controller or owner of some resources, usually based on the identity of the requester. Access Control Matrix is the conceptual DAC model proposed by [20]. Context-aware access control policy is defined in [21] where the policies change in accordance with the context or environment.

Due to the restrictive access of the resources by the agents we propose four types of resources to avoid agents accessing the resources. By explicitly modeling the different types of resources, we can avoid the system to enter into the undesirable state during the runtime. These resources are classified in the order of highly restrictive to least restrictive where the values can be altered frequently and can be accessed by every agent in the least restrictive resources.

1.3 Extended Tropos – A proposed approach

1.3.1 Security Constraints of Resources

The access control policies of the resources and secured agents are modeled in our extended Tropos representation such that it provides a way of access control of the resources. We also define a set of resources which are the defined resources, definable resources, static resources and synchronized resources.

Defined Resources: These resources are defined by the agents. The modification of these resources can be done only through the agents which created the definition.

Definable Resources: Definable resources are the set of resources where the resource values may be altered during the course of execution with the permissions from the agents which defined it.

Static Resources: The static resource values are common for all the goals/tasks; hence it is defined once and cannot be altered or changed at any time.

Synchronized Resources: In this synchronized resource, all the agents can share the resources and all the agents can update the resources simultaneously.

These types of resources provide the authorized access of the resources of agents because many security issues arise due to the lack of authorization of the resources to the appropriate agents.The representations of these resources provide a clear idea of how the requirements are modeled which are given in Figure3.

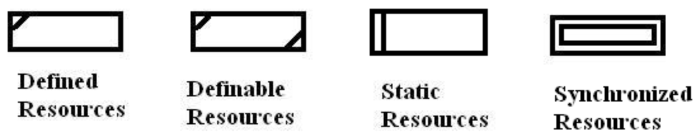

Figure 3. Representation of resources in our extended Tropos

In general, our approach considers the resources as the properties files, rules, definitions etc.

1.4 Temporal Factors and security modeling

We extend Tropos in fulfilling the resource securities and access policies of the resources in the distributed environment in which we consider the temporal properties of the goals (G) and tasks (T). Consider a set of tasks Ta, ta_i belongs to T where $0 \le i \le N$ and set of goals G, and g_j belongs to G where $0 \le j \le N$ .Let R be the set of resources which is being held by the resources and r_k belongs to R in which r_{g_i,t_i} is resource which is being held up by the task t_i and the goal g_j at the given instance.

There can be a single goal or task which holds the same resource at the same instance of time; hence the temporal ordering of the goals plays a vital role if the resources are being updated frequently.

Hence our approach concentrates more on the temporal ordering of the goals and tasks towards holding of the resources.

Definite Time Interval: A time interval of non infinite lengths are defined $t_{\min}$ and $t_{\max}$ where

$$0 \le t_{\min}, t_{\max} \le N$$

Relative Time Interval: This defines the relative time between the two goals, tasks or resources due to various constraints.

$t_{before}(g_j or\ ta_j)$ or $t_{after}(g_j or\ ta_j)$

Absolute time constraint: An absolute time constraint is a constraint of the form T **OP** Date, where T is a time variable, Date is a date and OP$\in$ {at, after, after or at, before, before or at} (with their intuitive meaning).

1.5 Temporal Constraints in ABDTS

In ABDTS, the total Storage Time (ST) required for storing content in a distributed system at a reticular remote location includes the processing time in the local node, network latency and the processing time of the remote node. Consider a failure occurs in the remote due to the Disk failure or any hardware failures, the storage time becomes infinity and the file cannot be accessed in the system. To avoid such conditions we set the maximum Time limit ST_{max} required to store a particular content in the distributed system. If an agent fails to

store content such as audio/video within the ST_{max}, the agents will consider that the storage failure is due to some failure in the system or network hence it will search for the other alternate solution.
In our system the Storage time (ST), access time (AT), replication time (RT), Look up time (LT) which is described falls under the category of defined time interval.
The Access Time (AT) is defined as the time required for an agent to access the resources for a definite amount of time. This access time is given by another agent which is AT_{min} and $AT_{max.}$ AT_{min} and AT_{max} definesthe minimum and the maximum time for an agent up to which it can hold the resources. In this Access time agents can modify the values of the resourcesand hence it is considered as the permissions given for a particular agent to modify.The time required for an agent to retrieve the resources such as audio/video from the remote location is defined as the Retrieval Time (RT).The retrieval time RT includes the lookup time (LT) which is the time required for an agent to search the directory for a particular entry. This LT may be further classified into Naming directory LT (NDLT), Location directory (LDLT) and Storage directory (SDLT).The Storage time (ST) also includes the additional time of NDLT, LDLT and SDLT which is the maximum time with which the agents can lookup the appropriate directories.
The Time interval is relative if initial instant and final instant are defined in function of another instant.The Lookup time is relative to the Retrieval Time, as it should occur before the retrieval and storing of the contents. In ABDTS the replication of the contents are done periodically which comes under the category of the Absolute time. The Replication time (REPT) is defined as the time after which the replication of the contents has to be done.But this replication or restoration of the replicated copies canoccur after some definite time or at a particular time or date.

IV. ABDTS Development

In this section we discuss the development of Agent based distributed tutoring system by applying formalTropos and extension of Tropos proposed by us.

1.6 Early-requirements

The distributed modeling starts from the early-requirements, analyzing the goals in organizational perspective. In our approach we take the user who is going to access the system, store/retrieve/edit the content but the end-user has no knowledge about where the contents are being stored. In Figure4 the resources are modeled as the synchronized resources in which the tutoring agents can change the data and meta-data of the resources; hence it is defined as the synchronized resources. The synchronized resources are the resources which can be shared by all the agents and can be modified during the execution time. In Figure.4 the audio and video resources are modeled as synchronized resources such that it can be accessed, modified by the Access Content, Store Content, Edit/Save Content Agents.

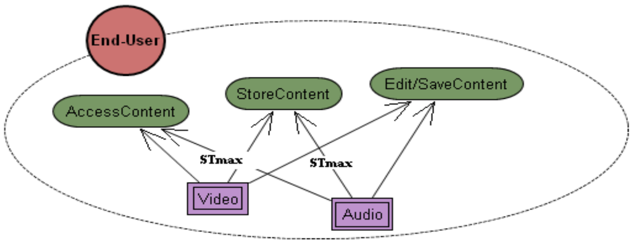

Figure4. Goal Diagram-Early Requirements

1.7 Late Requirements

In the late requirements, we introduce the system-to-be where the system is being modeled with its different goals and tasks. Figure5shows the system-to-be requirements where the different goals such as retrieve contents, store contents, replicate contents are modeled.

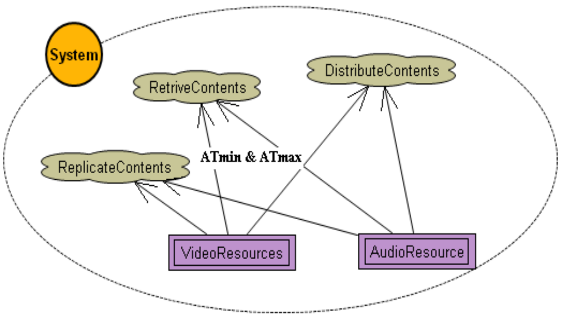

Figure5. Late Requirements- System-to-be

In our approach we defined different agents, and modeled their goals, roles and goal dependency using the Tropos methodology. Figure5depicts the storage Agent which has the goal of storing the contents across the distributed environment. It can also make decisions whether to distribute the contents or store the contents in the local disk. This is decided by many property files such as the Access control Policy, Location directory, Storage directory, Naming Directory etc. In Figure6, the location directory is modeled as the definable resources because the location directory contains the location of the resource and this location changes from time to time. The naming directory is also modeled as the definable resource as it also changes in course of time and it can be modified only by the agents which created it. The access control policy is a defined resource as the directory is not changed in the entire system.
The replication of the resources in the distributed environment is decided by the replication policy which is defined in the system. The replication algorithm in [4] is used for our replication strategy. Figure 7 explains the replication agent in which replication agent goals, strategies, resources are described. The optimal replication is a soft-goal, which can be measured by the appropriate parameters. In Figure 6 the naming policy and the replication policy are modeled as the definable resources as they can be only modified by the respective agent which created it.
Once the replication policy has been changed, it results in the degradation of the entire system. The replication policy can also be set to infinite number of copies which may collapse the distributed system.

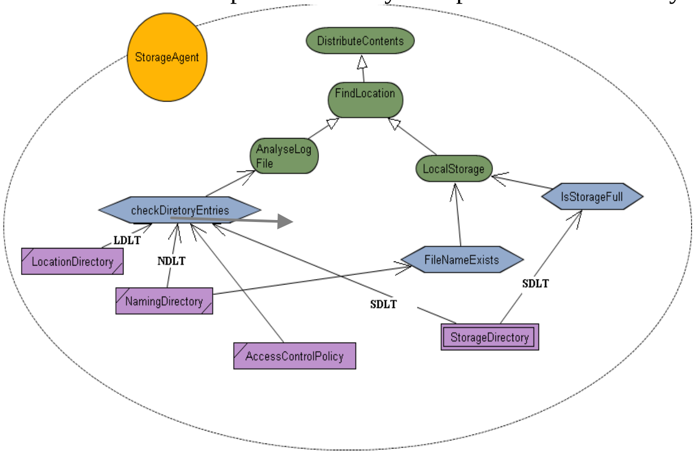

Figure6. Late Requirements- Storage Agent

The check directory Entries task requires the LDLT time for checking all the entries in the directory. It is the permitted time for an agent to check the entries in the directory, if the agents look up time exceeds the handle to this directory is released.

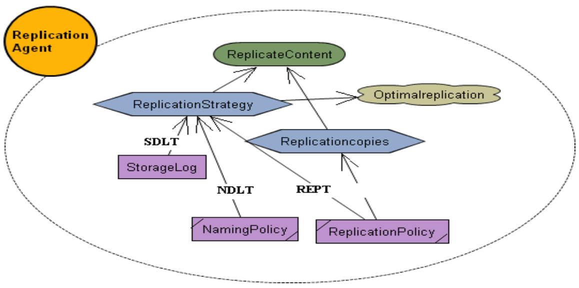

Figure7. Late Requirements- Replication Agent

V. Agent and Roles in ABDTS

We define Agents such as the resource allocation agent, Distributed Naming agent, Replication Agent, synchronization agent, Resource retrieval Agent, etc. In Table 1, the roles played by the agents, instances of these agents in the run-time, dependent agents and life time of these agents are elaborated. The roles of the agents are the functional specification assigned to the agents and the number of instances of the agents clarifies the total no of agent's instance created in the run time and how these instances are created. The Dependent agents specify the agents that are dependent for the successful execution of the goals. The life time of the agents is the time difference between the creation time and the deletion time of the agents in the system. Thus Table 1 specifies the agent's roles, instances dependent agents and the lifetime of agents which leads to clear understanding of the agent's behavior in the system.

Table 1. Agents and Rolesidentified in ABDTS using extended Tropos

Agent Name	**Agents and Roles**			
	Role	***Number of instances***	***Dependent Agents***	***Life Time***
Storage Agent	Stores the Content in the appropriate nodes	Number of request for the storage in the resources	Replication Agent	Up to Storage Succeed
Replication Agent	Replicates the given resource based on the policies and naming structure	Depends on the system	Policy Analyzing Agent	Replication succeed
Resource Retrieval Agent	Retrives the entire resource and re-produces to the end-user	Number of resource retrieval requests	Resource Consolidation Agent	Retrieval and consolidation time
Naming agent	Naming agent provides the naming conversion to the users. Local name and the distributed naming transparency levels are to be processed	Local naming agent, global naming agent, replication naming agent	Policy Analyzing Agent	Entire session of the end-user
Policy Analyzing Agent	Analyses the policy and the directory and provides authorization	On-demand	Replication Agent, Naming Agent	On-Demand
Tutor Agent	Analyses the Student's profile, interest and provide contents	Number of Learners	Learner Agent	Learner's session
Learner Agent	View all the apaptive contents	Number of Users	Tutor Agent	Entire Session

VI. UML vs.Extended Troposand AUML vs. Extended Tropos

UML is a universally accepted industrial standard modeling the software system which usually consists of the functional requirement analysis through the use-case diagram, sequence diagram, activity diagram, class diagram deployment diagram etc. These diagrams help in designing the software at different levels of abstraction.AUML is the extension of the UML modeling, which defines the meta-model for the agents roles and responsibilities through class diagram, the different activities of the agents are modeled through the activity diagram.

UMLhas proven its strength in the last decade but when the agent oriented software is concerned, UML lacks in adaptive requirement analysis, goal modeling, dependencies between goals and the security aspects of the resources. In both UML and AUML there is a lack in the requirement elicitation and modeling phases.Table 2 describes the comparison between UML, AUML and Tropos in the aspects of the requirement analysis and software development.

Table 2. Comparison of UML, AUML and extendedTropos

Properties	UML	Extended Tropos
Adaptive Requirement Modeling	No	Yes
Identification of Goals and tasks	No	Yes through Hard Goals and soft goals
Refinement of Goals	No	Yes
Temporal constraints	No	Yes
Resource Security Modeling	No	Yes (In our extended Tropos)
Identification of the Agents	In the Use-case Diagram but as Actors	Both Early Requirement analysis and Late Requirement Analysis
Agent roles and Responsibilities	No	Yes

From the Table 2 we can infer that the Tropos posses a greater stability in the development of adaptive systems. Traditional modeling tools like UML and AUML are not that much appropriate for modeling agent oriented system because of the following reasons, it is not possible to model the requirements and also agents are derived from its goals and tasks it is not possible to model the goal based requirements in the formal UML approach. Moreover goals/tasks require the segregation into sub-goals/sub-tasks and AND/OR decomposition of goals/tasks, but UML has "includes and extends" relationship in the use-case diagram which cannot model the goals and tasks. AUML has a capability to model the capabilities, communication language, ontology, protocol and properties of agents. AUML also lacks in modeling the requirement of agents, goals, tasks, dependency between goals/tasks. And also there is always a lack of modeling of the distributed system; hence for such development of agent oriented distributed systems, Tropos and extended Tropos provides an appropriate modeling Technique.

VII. Conclusion

In this work we have extended the Tropos methodology by considering the security of the resources and the temporal constraints. Agent Based Distributed Tutoring System is modeled in a systematic way by applying the extended approach, which makes the system stable. As the agents based systems are scalable future enhancements in the system consume less time and the designers need not work from the initial phase. The proposed extension of Tropos is not only suitable to model Agent based tutoring systembut also suitable for the domains such as Telecom and Banking industry where security plays a vital role.

Reference

[1]. Henderson-Sellers, B., Giorgini, P. (eds.): Agent-Oriented Methodologies. Idea Group Inc. (2005)
[2]. Federico Bergenti, M.-P.G., Zambonelli, and F. (eds.): Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook. Springer, Heidelberg (2004)
[3]. P. Bresciani, P. Giorgini, F. Ginuchiglia, J. Mylopoulos, and A. Perini, "TROPOS: An Agent-Oriented Software Development Methodology", In International Journal of Autonomous Agents and Multi Agent Systems, 8(3):203–236, 2004
[4]. JoséM.Pérez,FélixGarcía-Carballeira, "Branch replication scheme: A new model for data replication in large scale data grids", Future Generation Computer Systems vol.26(2010)
[5]. P.Kokkinos,K.Christodoulopoulos,E.Varvarigos, "Efficient data consolidation in grid networks and performance analysis" in FutureGenerationComputerSystems, vol 27, 2011.
[6]. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Jnl. of Information and Software Technology 38(4), 275–280 (1996)
[7]. Badjonski, M., Ivanovic, M., Budimac, Z.: Intelligent Tutoring System as Multi-agent System. In: IEEE International Conference on Intelligent Processing Systems, pp. 871–875 (1997)
[8]. Chang, Y.H., Lu, T.Y., Fang, R.J.: An Adaptive E-learning System Based on Intelligent agents. In: 6th Conference on WSEAS International Conference on Applied Computer Science, pp. 200–205 (2007)
[9]. Nargesian, F., Nazemi, E.: An Agent-Based Architecture with Centralized Management for a Distance Learning System. In: International Conference on Computational Intelligence and Multimedia Applications (2007)
[10]. Jade-Java Agent Development Framework, http://jade.tilab.com/
[11]. Foundation for Intelligent Physical Agents, http://www.fipa.org/
[12]. Volha Bryl, Paola Mello, et al, "B-Tropos Agent-oriented requirements engineering meets computational logic for declarative business process modeling and verification", in proc 8th Workshop on Computational Logic in Multi-Agent Systems (CLIMA-VIII), 2007.
[13]. Hadar, I, Reinhartz-Berger, I, Kuflik, T., et al," An empirical study of Requirements Model understanding: Use Case vs. Tropos models", in proc 25th ACM Symposium on Applied Computing, 2010.
[14]. Siena, A., Mylopoulos, J., et al, "Designing Law-Compliant Software Requirements", in proc. ER 2009.
[15]. Marco Montali, Paolo Torroni, et al, "Engineering and Verifying Agent-Oriented Requirements augmented by Business Constraints with B-Tropos", in Auton Agent Multi-Agent Systems, vol 23, pp. 193-223, 2011
[16]. Borselius N (2002) Mobile agent security. Electron Commun Eng J 14(5):211–218
[17]. Youna Jung, Minsoo Kim et al., "A survey of security issue in multi-agent systems", Artif Intell Rev, 2011
[18]. Beydoun G, Gonzalez-Perez C, Henderson-Sellers B, Low G , " Developing and evaluating a generic metamodel for MAS work products". In: Garcia A (ed) et al Software engineering for multi-agent systems IV, LNCS 3914. Springer, Berlin, pp. 126–142,2006
[19]. Beydoun G, Low G, Mouratidis H, Henderson-Sellers B ,"A security-aware metamodel for multi-agent systems (MAS)". Inf Softw Technol 51:832–845
[20]. Lampson BW (1974) Protection. SIGOPS Oper Syst Rev 8(1):18–24, 1974
[21]. ZhangG, ParasharM(2004)," Context-aware dynamic access control for pervasive applications", In: Proceedings of communication networks and distributed systems modeling and simulation conference, 2004
[22]. Vivek tiwari, S.k. Lenka, Shailendra Gupta, Renu Tiwari, "Performance evaluation of Java Remote Method Invocation and Mobile Agent Techniques in context of Distributed Environment", in Proc of International conference on Networking and Information Technology,pp.303-308, 2010
[23]. Ciprian Dobre, Ramiro Voicu, Adrian Muraru, Iosif C. Legrand, "A Distributed Agent Based System to Control and Coordinate Large Scale Data Transfers", in Proc. of 16th International Conference on Control Systems and Computer Science (CSCS-16), pp. 64-68, Bucharest, Romania, 2011
[24]. lacob (Ciobanu) Nicoleta – Magdalena ," The use of distributed databases in e-learning systems", Procedia - Social and Behavioral Sciences, Volume 15, pp. 2673-2677, 2011
[25]. Lawrence Cabac and Daniel Moldt," Formal Semantics for AUML Agent Interaction Protocol Diagrams", Lecture Notes in Computer Science, 2005, Volume 3382/2005,pp. 47-61, 2005
[26]. Kiran Mishra and R.B. Mishra," Multiagent Based Selection of Tutor-Subject-Student Paradigm in an Intelligent Tutoring System", in International Journal of Intelligent Information Technologies, Volume 6, Issue 1. 2010
[27]. Guilherme Bittencourt, Janette Cardoso,Luciana B. Frigo, Adriana Postal, Eliane Pozzebon, "Mathtutor: A Multi-Agent Intelligent Tutoring System", IFIP International Federation for Information Processing, 2004, Volume 154, Artificial Intelligence Applications and Innovations, pp. 231-242, 2004
[28]. Xianye Li, Fahui Ma, Shijun Zhong, Lin Tang and Zhongwei Han, "Research on Virtual Experiment Intelligent Tutoring System Based on Multi-agent", in Lecture Notes in Computer Science, Volume 6249, 2010
[29]. Xianli Huang; Dongmei Luo, "Applying interface agent in intelligent cooperative learning system", in proc. 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), pp. 4564 – 4567, 2011

[30]. Kumar, R.; Rosé , C.P., "Architecture for Building Conversational Agents that Support Collaborative Learning", in IEEE Transactions on Learning Technologies, vol 4, issue 1, pp. 21-34, 2011
[31]. http://nptel.iitm.ac.in/
[32]. http://ocw.mit.edu/
[33]. http://www.youtube.com