An Autonomous Provenance Tracking System for Collaborative Environment

Syed Imran Jami and Zubair A. Shaikh

Department of Computer Science, National University of Computer & Emerging Sciences (FAST-NU), On National Highway, Karachi Pakistan.

Abstract: Provenance is the process of recording the origin of digital object. These objects include data, documents and chunk of information. Provenance Systems are now very much in use in different domains however they are not working autonomously as a separate entity. This makes them heavily dependent on target applications. We, in this work proposed an autonomous provenance tracking system by separating the tracker as separate layer from the target application. The problem of autonomy is solved by employing standard and interoperable multi agent systems where agents are interacting with the application agents for tracking the log of usage. The target application is the set of collaborative applications where users are sharing their thoughts. The autonomous provenance tracker is tested on Whiteboard collaborative application in an effective and efficient way. We have used FIPA compliant JADE for multi agent based platform that are programmed for recording the usage of Whiteboard.

Key words: Provenance, Autonomous, Tracking System, Collaborative Environment

INTRODUCTION

The web based collaborative tools are nowadays used for sharing the thoughts in an automated way. They are rapidly changing the world of E-Mails and Chat based applications for collaboration. Collaborators, instead of generating E-Mails and Chat logs, are now generating digital objects. The digital objects include data, documents and chunk of information. Many of these digital objects are valuable artifacts that help in creating, deriving and extending other collaborations. The availability of huge set of objects has created concern about their validity and authenticity. Provenance is a metadata of target object that contains data about its usage, creators, set of transformation processes, ancestors’ etc. Provenance aware collaborative applications provide this metadata of information generated from collaboration.

Collaboration as a process is defined in (Jacob Gub, 2011) as ‘individuals working together in a coordinated fashion, towards a common goal’. The related applications provide the platform for teams working together over geographic distances. With the growing usage of Virtual Organizations (Criado et al., 2009) in businesses, these applications require tools that help in communication among the collaborators for problem solving.

The survey in (Jacob Gub, 2011) provides number of tools available for communication and collaboration. This includes popular tools like instant messaging, video conferencing applications, social networking websites etc. Some other complex tools include project management and flowchart based designing tools. Social tools for collaboration include Skype, Facebook, LinkedIn etc.

Apart from Communication tools, Collaborative applications can also provide tools for Conferencing and Collaboration management. The details in (Hall, Calvin, 2011) provides the comparison among these mechanisms by describing Communication as ‘unstructured interchange of information’ (for example phone call or chat discussion) while conferencing refers to ‘interactive work toward a shared goal’ (for example brainstorming) and Co-ordination refers to ‘complex interdependent work toward a shared goal’.

In this work we are providing an autonomous and automated provenance tracking system for keeping the track of communication among the collaborators. Our main hypothesis is to keep the provenance mechanism separate from the applications. The autonomous provenance tracking can work with any application without requiring intervention from users. To test this hypothesis we selected collaborative application as our testbed. Under this collaborative platform we are dealing with the communication aspect by implementing the whiteboard (Jacob Gub, 2011) as our collaborative application. In this collaborative application, user can
perform the operation of create, edit and comment on whiteboard. When the user log on to the application an agent is fired who is responsible for tracking all the actions performed by that user that are stored as provenance metadata.

The next subsections provide details about our whiteboard application that is interacting with our Provenance Agents. This work is an extension of our earlier works presented in (Zubair Ahmed Shaikh; Syed Imran Jami et al., 2008) where we developed the tracker using Aglet (Lange and Mitsuru, 1998) framework as Agent platform. Aglet based Provenance Agents were mobile in nature thus leads to extra communication overhead. In this work we implemented Provenance Agents without mobility feature which leads to good efficiency with lesser overhead.

Related Work:

The topic of Provenance Systems becomes prominent in recent years. Many labs and researchers are working towards these systems. However, the current focus of most of them is towards its representation structure. The community is nowadays more focused towards common representation model and it recently proposes Open Provenance Model [ref]. Very few systems are available for tracking of provenance record. Out of these systems, some frameworks propose to assign the tracking task in workflows for example the approach used by (CHEBOTKO, et al., 2007; SIMMHAN et al., 2006) while some others develop stand alone applications that work as processes that are responsible to create and manage provenance for example the work of (BUNEMAN et al., 2006; FOSTER et al., 2002) and (GROTH et al., 2005). The other category suggests that operating system routines should be programmed to record the provenance (MUNISWAMY-REDDY et al., 2006).

In all these systems provenance tracking is embedded in the target applications. For robustness and complete automation, an autonomous tracking feature is required. One of the promising approaches in this regard is the use of agents to record the provenance metadata. On accessing or using the digital artifact by the user, the basic idea is that the agent will record the log information. These agents can work as an underlying architecture in layered model where workflows, applications and operating systems are relieved from the recording tasks. (Kiőr et al., 2006) uses agents to record provenance in distributed environment. They applied this concept on health care system provenance aware agents in it. They adapted PASOA architecture (described in (GROTH et al., 2005)) for recording provenance in health system. This system however works only for the unique environment of health care system.

Proposed Approach:

In (Fabio Bellifemine), Agents are defined as ‘sophisticated computer programs that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems’. Multi-agent system (MAS) provides a loosely coupled platform for collaborative set of software agents that interact with each other to solve problems. Such problems require Distributed AI based techniques since individual capacities or knowledge alone cannot solve them. Examples include the logic used in heuristic games (for example chess, chequer or the football), business scheduler (travel, hospital etc) in which different agents are required to represent the preferences of different people. Moreover, it is shown in many researches including our earlier paper in (Zubair Ahmed Shaikh) that use of multiple agents speed up the tasks by providing a software based method for parallel computation. Similarly due to its autonomous nature robustness is another benefit due to which they are not vulnerable with the target application. Scalability is another benefit shown by researchers. As compared to monolithic and integrated system it is easier to add new agents to a multi agent system with the growing number of operations along with users.

Thus the advantages of efficiency, robustness and scalability of Multi Agent Systems due to their autonomous nature help our provenance system to achieve these benefits. We considered these aspects of special importance because collaborative applications work in distributed and open environment.

In this work we have proposed a layered approach for provenance tracking as shown in figure 1. This makes our provenance tracking independent (autonomous) to the application. The top layer is the application layer in which we developed white board collaborative application. The bottom layer is provenance layer in which the metadata is recorded into the log file. The middle layer is working as an interface between the collaborative application and provenance agents. The next subsections detail each of these layers from design and implementation perspective.

Application Layer- Whiteboard:

Whiteboard is a popular web based collaborative application used for collaboration. Figure 2 shows the
screenshot of our developed application in which user first initiates the Whiteboard session who is assigned its owner. The owner can invite other fellows to collaborate on that whiteboard for any discussion. The collaborators can comment and edit on it to share their thoughts. These discussions are saved for later use.

**Fig. 1:** Layered Architecture of Proposed Design

**Fig. 2:** Screenshot of Whiteboard Collaborative Application

In this application the previous chunk of information is available and can be discussed later at any time. This application is later developed as provenance aware through which we can answer the complex queries about users’ actions and contribution along with temporal data

**Agent layer–JADE(Java Agent Development Environment):**

We have used JADE to implement our agent layer. JADE (Java Agent Development Environment) is a software framework for developing the multiagent applications in compliance with the FIPA standards and specifications. It is an open source and FIPA compliant specifications. FIPA (foundation for intelligent physical agents) is a standard body that describes the rules and specification for Multi Agent Framework. There are three components of JADE platform as described in (Fabio Bellifemine).

- The Agent Management System (AMS) is the agent that enjoys supervisory control over the platform; it is responsible for maintaining the directory of resident agents and for handling their life cycle.
- The Agent communication channel (ACC) as the name implies, is the channel that provides the contact between the agents within and outside the agent platform. It ensures orderly, reliable message routing service to agents.
- The Directory Facilitator (DF) is the agent that provides the yellow pages services to the agent platform.

FIPA describes ATP (Agent Transfer Protocol) as the protocol for agents to exchange messages with each other (Fabio Bellifemine). JADE uses ACL (Agent Communication Language) to implement ATP.

Figure 3 shows the screenshot of JADE platform that shows the working of AMS, ACC and DF components. These components are by default activated at the start up of the platform. They control the distributed agents in its platform. It provides Java API to send/receive messages to/from agents. ACL messages are treated as ordinary java objects (Fabio Bellifemine).
Fig. 3: JADE Platform

**Provenance layer- Log File:**

This layer is responsible for tracking the log of information interchange among the collaborators. The logs are stored as metadata of the whiteboard session. It keeps the information about the discussion on whiteboard for example:

- Who (user) created the whiteboard?
- At what date and time he created it?
- Who (user(s)) edit the whiteboard at particular date & time?
- Which users comment on the particular document?
- Who delete the contents at particular date and time?
- What set of actions performed on this discussion after reaching to some conclusion?
- What set of actions this (for example malicious) user performs on this collaboration?

The list is exhaustive and by proper querying mechanism user can get the answer of these complex queries.

![Diagram of JADE Platform](image)

Fig. 4: Provenance aware Interaction among Collaborators

Figure 4 describes the interaction of three layers described above. The users are interacting on the application layer using Whiteboard. The agent layer is tracking the users’ actions on the application like when the user creates, edit comment on the whiteboard and the time of all the respective events. The tracked data is stored in the provenance layer in a log file.
RESULTS AND DISCUSSION

The complete integrated system is tested for correctness using black box testing technique. We have tested the features and functions of the application as per the specifications. Details are out of scope of this paper. We have validated and verified the system, tested whether the specified functions are implemented and whether these functions are behaving correctly as per the expected behavior.

The system is tested for scalability and efficiency in a distributed setting. The system is loaded with multiple provenance agents and behavior is observed in terms of its response time, performance deviation and deterioration. Secondly we have also measured and analyzed the performance of the system under multiple users and agents interacting with the system. The following graph best describes the performance of the system.

![Graph showing timing of different modules of our collaborative application using Agents](image)

**Fig. 5:** Graph showing timing of different modules of our collaborative application using Agents

The graph above in figure 5 describes the performance of the system under multiple agents interacting on the system and performing multiple actions. The vertical axis indicates the time in seconds and the horizontal axis represents the number of agents. The different color indicates the different actions being performed by the user on the application. As it is clearly noted that the time of the first agent is relatively higher than other agents’ simultaneous working and it is 28 seconds performing all the five operations on the system. This may be due to the time taken by the server to initialize its services and the internal operations of the JADE platform per se.

As we increase the number of agents it is small increase in response time with the average time is in the range of 20 to 23 seconds.

![Graph showing timing when multiple agents are collaboratively working on single task.](image)

**Fig. 6:** Graph showing timing when multiple agents are collaboratively working on single task.
Figure 6 shows the performance of the system under multiple agents performing single activity. The four lines are of the actions performed on the whiteboard. It is noted that the first agent took the highest time in seconds (8 seconds) to create an original whiteboard and then the time decreases drastically in creating in the existence of two parallel agents and afterwards the time remains at the upper threshold of 4 seconds till the 10 agents running at the same time.

The time to save the whiteboard lies in the range between 4 to 5 seconds. Similarly the time for comments lies in the range of 3 to 4 seconds.

The small impact of increasing number of agents show the tolerance of multi agent based architecture with respect to scalability and efficiency as described in section 3.

![Single Activity Graph](image)

**Fig. 7:** shows the overhead time required to track the data in the log file

Figure 7 shows the overhead time required to save the data in the log file in the presence of multiple agents. For the sake of simplicity we have traced the time required to create the whiteboard with provenance data being stored in the log file. This graph shows the overhead of provenance architecture over the collaborative application. When one agent is active, the time required to save the provenance data of the user is 2.8 seconds and this time is reduced in the creation of latter agents till the 9th agent is erected. The 10th agent shows the increase in time for saving the data.

**Conclusion:**

In this work we have developed Whiteboard as test bed application. Due to autonomous provenance layer any application can work with our provenance agent through Agent Layer. The change in application will not have any effect on the agent layer.

The major contribution of this paper is the proposition of provenance framework for tracking the metadata in distributed environment in which network resources are open scattered across the globe. To meet these requirements we propose a provenance tracking system for a collaborative system that can work in efficient way with good scalability. Our results in figure 5-7 show that these targets are achieved using integration of Provenance Tracking with Multi Agent Systems.

In our earlier work (Zubair Ahmed Shaikh) and (Syed Imran Jami and Zubair A. Shaikh, 2008) we use Aglets (Lange and Mitsuru, 1998) in our architecture for this problem. However, with the mobility feature of Aglets, the ‘heavy’ agents passing among nodes as messages (Lange and Mitsuru, 1998) result in extra communication overhead. This overhead is remarkably reduced using stationary JADE based agent system. The use of multiple agents for each set of actions also achieves scalability by separating the concerns among different agents.

**REFERENCES**


1673


