Design Proposal Report

Intelligent Digital Forensics Agent

1. Executive Summary

Enron Corporation's legal team is working on resolving recent press rumours regarding illegal activity within the company. RMRK Consultants are tasked with helping the team in analysing their internal personnel communication, to identify any individuals that may be behind the alleged illegal activities.

This report discusses the design, testing, and suggested implementation of an agent-based model that aids the legal team in conducting internal digital forensics. Utilising a sample email dataset, the team can enter suspect names and find out relevant data (such as terms used within the emails, and their frequency; top senders of the emails in question, and provide a match for any particular query). This report further discusses permissions and protocols required by the model design, and the languages/modules that will be utilised accordingly. Testing and implementation metrics have also been discussed herein.

A follow-up report will be provided with the results of the said model's implementation using the actual dataset provided by the legal team, to identify if the press claims are true or falsified.

2. Introduction

RMRK Consultants have been hired by Enron Corporation's legal team to provide an initial report, suggesting a digital forensics agent-based solution to the proposed problem.

3. Business Understanding

a. Business Objectives

This report discusses an agent-based model to help Enron analyse their internal personnel communication to identify the individuals involved in suspected illegal activity.

b. Situation Assessment

Enron's company image is influenced by recent rumours, and the legal team wants to ensure that the internal personnel held responsible are identified and course-correction can happen. The investigators have access to an initial list of suspects, and want to identify any additional suspects who may be involved.

c. Dataset

A sample dataset has been used to test this design (Cohen, 2015). The dataset for a comprehensive report and analysis shall be obtained from Enron’s legal team.

4. Design approaches

An agent-based solution is appropriate for the task because the environment is uncertain (Jennings and Wooldridge, 1998). The design approach will be based on the Agent-Oriented Analysis and Design methodology (Wooldridge et al. 1999).

Firstly, we will consider the agent roles. These will be related to their specific functions as given below (see Figures 1 and 2 for Unified Modeling Language (UML) representations):

- CoordinatorAgent - coordinating the search within the available email dataset (environment), using the initial evidence provided by the actor (such as suspect personnel names), and the developing feedback/evidence (such as resulting query matches, frequently used terms within the emails, and/or top senders of the said emails)
- SearchAgent - reading and searching the emails for a match to the initial evidence,

- AnalysisAgent - analysing the discovered data and reporting back to the coordinator, and
- CoordinatorAgent - saving and displaying the progress and results.

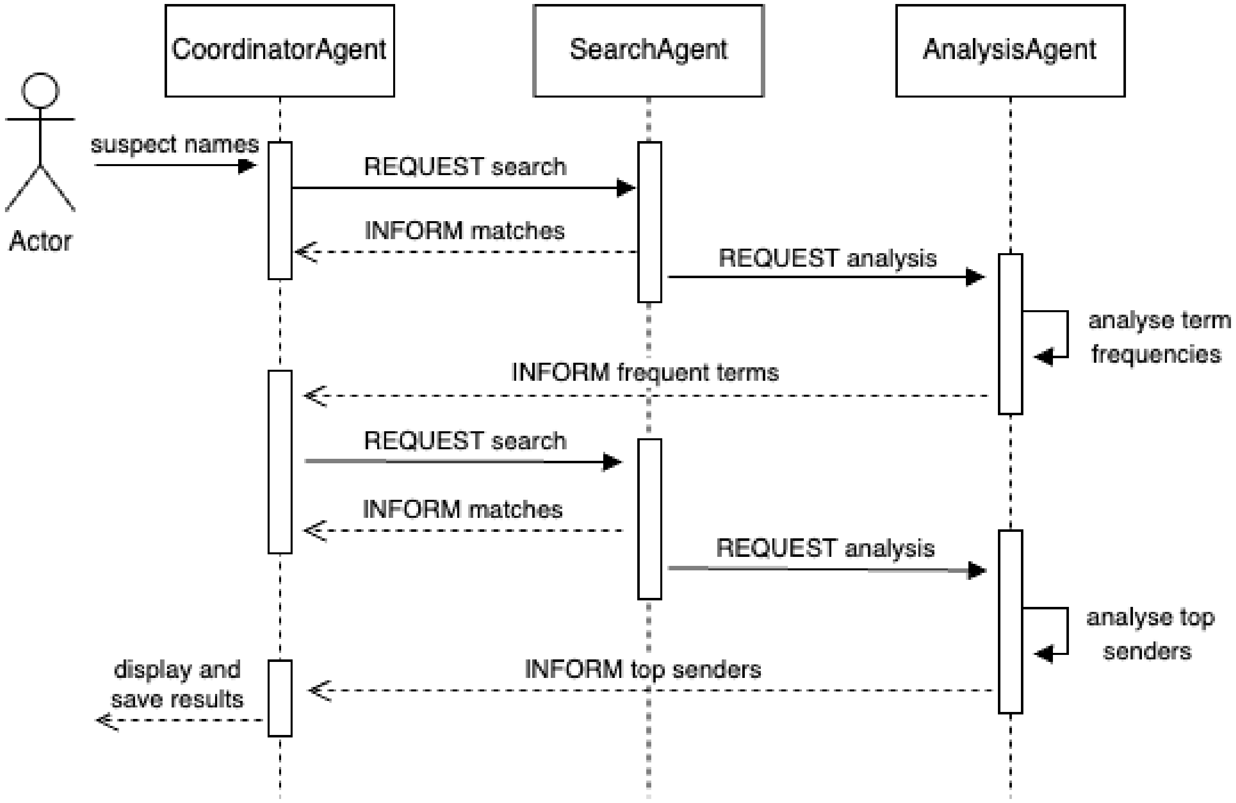

Figure 1. Sequence UML Diagram

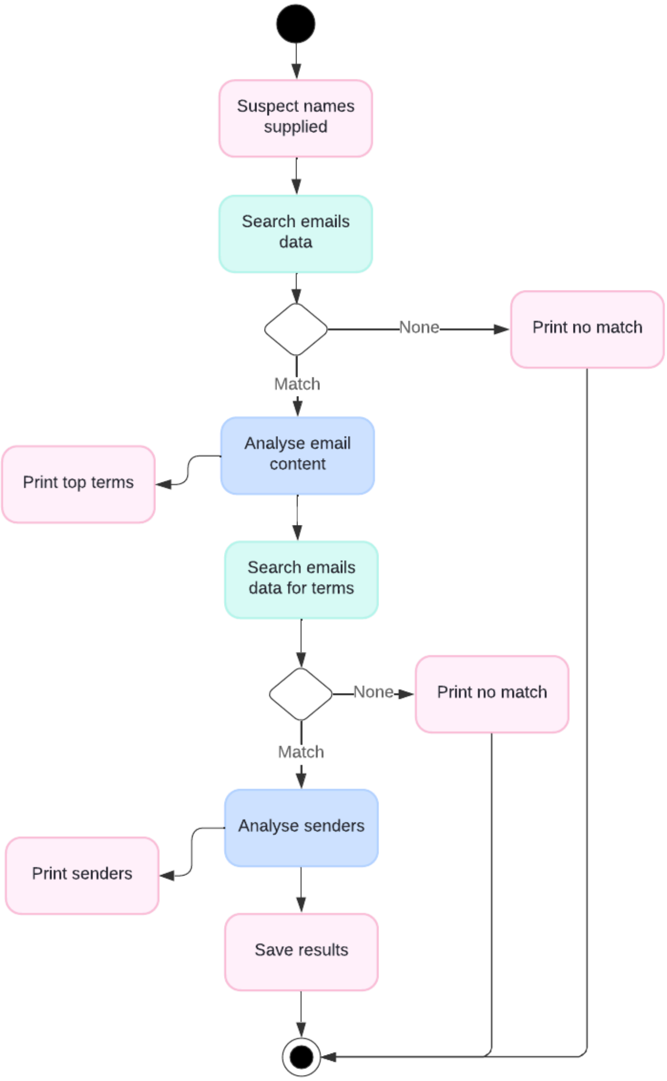

Figure 2. Activity UML diagram

Next, we will consider the permissions. The most important is the permission for SearchAgent to access the file system (herein, the email dataset).

Further, we will consider protocols. We will use an agent communication language, exploring Knowledge Query and Manipulation Language (KQML) (Finin, 1994), and the message parameters by the Foundation for Intelligent Physical Agents - Agent Communication Language (FIPA-ACL) (2001), with Knowledge Interchange Format (KIF) (Stanford, n.d.) as the content format. KIF is based on first-order logic. The advantage of FIPA-ACL is the unambiguous performative component. The pattern of interactions will be such that CoordinatorAgent will REQUEST SearchAgent to search for matches, and the latter will INFORM the former on the results. SearchAgent will REQUEST AnalysisAgent to analyse the results, and the latter will then INFORM CoordinatorAgent.

5. System Design & Libraries

Agents will be implemented as classes, with methods for the key tasks of each agent. Agents will run concurrently (using the threading python module). Asynchronous systems pose challenges, for example the management of shared state and memory. To overcome this, message passing will be implemented as a queue (using the queue python module).

The activity will be as follows:

- The CoordinatorAgent will read a text file containing the initial evidence: suspect names.
- Using the os python module, the SearchAgent will ‘sense’ the environment (file system) and identify email files. It will read the files and search for matches to the suspect names. It will aggregate the matches, sending the results to AnalysisAgent, and informing CoordinatorAgent in the process.

- The AnalysisAgent will analyse the content using NLP such as n-grams (python module NLTK), reporting the results to CoordinatorAgent.
- Then, the coordinator will REQUEST another search of the emails for the top terms.
- The resulting content will be analysed by AnalysisAgent, and the result (most frequent senders) will be reported to the CoordinatorAgent.

6. Testing and Implementation

Each agent method will be unit tested. In particular, the SearchAgent will be tested with the sample data, copied into a test filesystem. As the communication protocol is essential, each agent will be tested to make sure they are conducting the required KQML communications (REQUEST, INFORM). The AnalysisAgent will be tested with the sample content. The unittest python module will be run from the tests/directory in the project root.

The implementation plan offered by RMRK consultants for Enron's legal team will include three steps: user training, monitoring and support, and feedback and iteration of the functionality.

7. Conclusion

This report discusses an agent-based model (best route suggested due to the uncertain environment) to help Enron's legal team (actor herein) in identifying company individuals involved in suspected illegal activity. The model is designed for the actor’s ease of use in conducting internal digital forensics. Utilising a particular email dataset, serving as the environment for this model, the actor can enter suspect names and find out relevant data (such as terms used within the emails, and their frequency; top senders of the emails in question, and provide a match for any particular query). The success of this model depends on adherence to needed permissions and protocols; however, the model will reflect user suggestions if it experiences a hurdle. Primarily, KQML, FIPA-ACL, and KIF will be utilised along with relevant Python libraries, however, other languages (as suited) may instead be incorporated. Finally, RMRK consultants will assist with testing and implementation of the actual dataset.

This model design can greatly assist the legal team in identifying internal personnel that may be behind the claims made by the press, and to further validate the truth in those claims.

8. References

Cohen, W. (2015) Enron Email Dataset. Available from: https://www.cs.cmu.edu/~enron/ [Accessed 28 February 2024].

Finin, T., Fritzson, R., McKay, D. & McEntire, R. (1994) ‘KQML as an agent communication language’ In: *Proceedings of the third international conference on Information and knowledge managementCIKM ’94* 456–463. Available from: https://dl.acm.org/doi/10.1145/191246.191322 [Accessed 29 February 2024].

Foundation for Intelligent Physical Agents (2001). FIPA Specifications.Available from: http://www.fipa.org/specs/fipa00061/index.html [Accessed 29 February 2024].

Jennings, N.R. & Wooldridge, M. (1998) Applications of Intelligent AgentsIn: Jennings, N.R. and Wooldridge, M.J. (eds.) *Agent Technology* Berlin, Heidelberg: Springer Berlin Heidelberg: 3–28. Available from: http://link.springer.com/10.1007/978-3-662-03678-5_1 [Accessed 29 February 2024].

Lucid Software Inc. (2024) Where seeing becomes doing. Available from: https://www.lucidchart.com/pages/ [Accessed 29 February 2024].

Python Software Foundation (2024). queue - A synchronised queue class. Available from: https://docs.python.org/3/library/queue.html#module-queue [Accessed 29 February 2024].

Python Software Foundation (2024). threading - Thread-based parallelism. Available from: https://docs.python.org/3/library/threading.html#module-threading [Accessed 29 February 2024].

Python Software Foundation (2024). os - Miscellaneous operating system interfaces. Available from: https://docs.python.org/3/library/os.html [Accessed 29 February 2024].

Python Software Foundation (2024). unittest - Unit testing framework. Available from: https://docs.python.org/3/library/unittest.html [Accessed 29 February 2024].

Stanford (n.d.) Knowledge Interchange Format (KIF). Available from: http://ksl.stanford.edu/knowledge-sharing/kif/ [Accessed 29 February 2024].

Wooldridge, M., Jennings, N.R. & Kinny, D. (1999) *A methodology for agent-oriented analysis and design* In: *Proceedings of the third annual conference on Autonomous Agents* AGENTS 99: 3rd International Conference on Autonomous Agents 69–76. Available from: https://dl.acm.org/doi/10.1145/301136.301165 [Accessed 29 February 2024].