Agent-Based Security Framework to Protect Integrated model for Decision Making in Disaster Management

Sohail Asghar and Ashir Ahmed

Clayton School of Information Technology, Monash University, Australia

Abstract

The security of disaster management systems is a serious concern because of the sensitive and complex nature of this domain. It is important; therefore, that disaster management experts and the development community deploy and make use of security implementation measures to protect these systems. Unfortunately, they often lack this knowledge, and their subsequent development efforts to cope with disaster mitigation activities are less effective than they might otherwise be. Therefore, it is highly important that a robust security framework must be imposed while developing an integrated model for decision support in disaster management. The purpose of this paper is to propose a multi-agent security framework in order to protect integrated model for decision support in disaster management. In this framework, a group of agents can carry out their tasks cooperatively in order to achieve an ultimate security goal for disaster management model which facilitates effective decision-making and potentially attempt to deal with the problem of security in disaster management systems in general, and disaster data in particular. These tasks consist of two parts: (i) an agent detects the potential attack; and (ii) and solver agent can either selects a security framework on the basis of pre-defined security integrity frameworks stored in a case-base or propose a run-time generated security integrity framework.

Keywords: Agents, Integrated Model, Security Frameworks, Decision Support, Disaster Management

Introduction

The number of users using disaster management models for decision support is increasing exponentially every day. As a consequence, the chances of the integrated model being attacked also increased dramatically. Therefore, such integrated models are so threatened by malicious attacks. The protection of these models is, therefore, more than useful, it is vital. This problem requires protecting these models and performing continuous monitoring of such models in real time execution events. The complexity of this problem makes the use of multi-agent system necessary. The existing solutions available for security and protection management are costly and complex. What required is a flexible, adaptable and affordable protected solution which provides greater autonomy. In this context, a multi-agent system can be designed to achieve the desired functionalities. Furthermore, agent technology is recognized as one of the fastest growing areas of modern research (Boudaoud and McCathieNevile 2002; Zaki and Sobh 2004).
The acronym **MILD DSS** stands for “A **M**ult-I-Layered Architecture for **D**isaster **M**anagement **D**ecision **S**upport **S**ystems”. Disaster Management can be characterized as a complex and highly multi-disciplinary area. The necessity of decision-making in disaster management often requires a decision support system to fulfil dynamic and rapidly changing decision needs. The MILD DSS is dedicated to supporting decision-making in disaster management and is designed for such applications. With the MILD DSS, a DSS developer can make use of the subroutines (decomposed from existing DSS models for disaster management) categorized on the basis of environmental characteristics, disaster dependency and common decision support needs. The goal of the MILD DSS is to produce an integrated model based on a disaster scenario.

The aim of this paper is to propose a multi-agent framework to protect an integrated model (which is the outcome of MILD DSS) for decision support in disaster management which was previously developed by (Asghar S., Alahakoon D. et al. 2006). The proposed framework consists of two main components: firstly, an agent which detects the potential attack; and secondly, the solver agent can either select a security framework on the basis of pre-defined security integrity frameworks stored in a case-base or propose a run-time generated security framework.

Furthermore, we have devised a framework of a group of individual named agents such that each agent cooperates and communicates with other agents in an environment. The attack detection knowledge is communicated to the solver agent (through communication and supervisor agent) in order to generate a security framework for the potential attack by making use of a case-base. Such combined knowledge and experience of the agent with the information coming from neighboring agents allows the agent to provide the optimal security framework against an attack. Therefore, we built a framework that can be used to provide a flexible integration of multi-agent technique in a classical environment to provide protection against attacks on integrated model for disaster management.

The paper is organized as follows. Section 2 presents the architecture of the MILD DSS and integrated model for decision support in disaster management. Section 3 proposes an agent-based framework to protect the integrated model and highlights the main components and agents of the framework. Section 4 describes an example scenario to illustrate the usefulness of the proposed security framework in order to protect the integrated model. Finally, Section 6 provides concluding remarks and future work.

**Integrated Model for Decision Support in Disaster Management**

In this section, we provide an overview of the integrated model generated by the MILD DSS to gain the understanding of the model which needs to be protected. The multidisciplinary nature and increasing complexity of the disaster management area has resulted in the increased use of Decision Support Systems (DSS). The need for decision-making in disaster management often requires a decision support system to fulfill the decision needs. Disaster Management Decision Support Systems (DMDSS) are dedicated to supporting decision-making in disaster management and are considered to be complex
applications possibly integrating advanced computing techniques and requiring state-of-the-art research and development efforts. Disaster management issues generally possess the following characteristics:

- a global perspective
- the uncertainty involved in decision-making
- a changing nature
- dynamic and multi-criteria decision support needs
- highly interdisciplinary

Due to such characteristics, decision support systems for disaster management have gained increased consideration by the research and scientific community. In this context, finding a possible solution to develop an effective and efficient DSS for disaster management is an appealing and challenging area of research. The literature reveals that currently the following problems are associated with the existing DMDSS:

1. It is evident from the literature that over the past two decades, a large number of decision support systems have been produced by various developers on the basis of specific needs in specific disaster types (Tufekci 1995; Cheng and HoKo 2002; Pourvakhshouri and Mansor 2003; Onate and Piazzese 2005). The problem in developing DMDSS is that a DSS developed on the basis of specific needs cannot be adapted to handle the various activities of a disaster with different characteristics. Furthermore, these systems cannot be readily modified with the changing needs of disaster management because they are developed to fulfil the specific and targeted needs of decision-making in a particular disaster.

2. The traditional DMDSS do not consider common and environmental factors across diverse disasters for effective decision-making in a new disaster situation.

3. In the disaster management domain, a large number of DSS models are available. Such models do not handle mutually exclusive problems and as such much overlap appears in their functionality. It also shows that most of the DMDSS overlaps the functionality for example, the simulation model for emergency evacuation (Pidd, De Silva et al. 1996), providing decision support for evacuation planning (Silva 2001), an integrated emergency model DSS for hurricane emergencies (Tufekci 1995), and regional evacuation modeling systems (Tufekci and Kisko 1991), are all designed for the same functionality (evacuation). Such overlapping of functionality justifies the need for integrating DSS in disaster management.

4. Since one disaster may trigger another (for example, a volcanic eruption may lead to forest fires), with the requirement of diverse DSS needs, it is quite possible that a number of different models may need to be used simultaneously.

5. There can be certain situations in which user needs can be fulfilled by executing one or more subroutines or modules. Therefore, when a new situation comes along, subroutines can be reused to provide the solution for a new problem. For example, the evacuation planning system (Pidd, De Silva et al. 1996) can be decomposed into modular subroutines; the decision support need in flood management “timely distribution of flood emergency information” can be fulfilled by execution of a few of the subroutine of evacuation DSS system.
instead of developing the complete DSS model from scratch. The existing DMDSS lacks such functionality.

Therefore, we raised the above-mentioned problems associated with the existing DMDSS based on the analysis of the previously developed systems. To address these problems, we need a framework which can provide the following functionalities:

- The framework should not intend to provide decision-making for a specific decision support need, for example, an evacuation modelling system which is capable of providing decision support to fulfil only the evacuation decision need during a disaster. Rather, we need a DMDSS which can be adaptive and tailored according to the dynamic needs of a disaster.

- The framework can easily reuse existing subroutines/DSS models/DSS submodels to overcome the overlapping problems. The justification for the reusability is an important issue which needs to be addressed due to the fact that different subroutines may serve the same decision support need in spite of the fact that there may be some common needs across different disasters. And more importantly, it is the common environment which becomes the basis for the reusability.

- Because of the large number of activities and issues within disaster management different DSS models were developed to fulfil the decision support needs. Therefore, multiple models are required in order to provide decision-making in disaster management. Model integration is suggested as a better alternative approach. Hence, we need a framework which can support model integration for effective decision-making.

We proposed an architecture which overcomes these limitations of the existing DMDSS and possesses the above-mentioned functionalities. The multi-layered architecture designed for such purposes is called MILD DSS which can improve the efficiency and reusability of traditional DSS developed for disaster management. The objective of the architecture was to enhance the effectiveness of the traditional DMDSS development process by integrating decomposed modular DMDSS subroutines focussed on a disaster scenario. Such subroutines are selected on the basis of the needs classification scheme (Asghar S., Alahakoon D. et al. 2006). The proposed architecture is adaptive through the selection of subroutines according to the changing needs of decision-makers in the process of decision-making. It produces an integrated model which provides support that fits current user’s decision support needs focussed on a disaster situation. Therefore, the functionality of the architecture can be summarized as facilitating the DSS developer by selecting functionally-independent modular subroutines from the disaster management domain to produce an integrated model based for a disaster scenario. It is emphasized that the integrated model is not based on a particular decision support need but is developed on the basis of dynamic decision needs of a disaster. The architecture of the MILD DSS is shown in Figure 1 which consists of three layers: independent, dependent and interface. The functionality of the layers and the components of the proposed architecture are not discussed here as this is beyond the scope of this paper. Further details of the architecture can be found in (Asghar S., Alahakoon D. et al. 2006).
A prototype was developed to validate the conceptual model and tested on the basis of a wildland fire case study known as the Hayman Fire (Graham 2003). The Hayman Fire is a good source of a large number of decision support needs which can be fed into the MILD DSS to generate a dynamic integrated model based on such decision support needs.

Figure 1: A Conceptual Architecture of MILD-DSS

In order to support the decision-making in the disaster management domain, the integrated model achieved using MILD DSS possesses the following characteristics:

- Extensibility, which in a DMDSS refers to the ability to generate new DMDSSs, for new applications by modifying or building upon previous DMDSSs. The obvious advantage of such capability comes in terms of minimal cost and reduced development efforts. We emphasize that MILD DSS is designed to serve the extensibility.

- Previously designed and developed DMDSS models are focused to serve a specific decision task. For example, some DSS models have been developed and are currently used for flood risk assessment and management such as RAMFLOOD (Onate and Piazzese 2005). But the MILD DSS is not designed to fulfil specific decision support needs and requirements of disaster management.
but focuses on generating an integrated model based on dynamic decision needs. Therefore, the MILD DSS generates modular subroutines (integrated model) required for decision tasks for different disaster scenario. The concept of an integrated model is not to develop a single DSS model that addresses all decision support needs, but to facilitate the design of such an integrated model.

- The knowledge base of the MILD DSS is composed of existing modular subroutines developed for DMDSS (but labelled and grouped according to needs classification scheme). The integrated model is generated using such subroutines for different applications eliminating the need for duplication and reducing the possibility of errors and inconsistencies.
- The focus of the MILD DSS is to propose an integrated model that effectively supports decision-making. The effectiveness comes with the use of modular DMDSS subroutines achieved on the basis of dynamic decision needs.
- The MILD DSS is easy to use and better addresses user decision support needs for a given scenario.
- The modular subroutine selection criteria, associated with the MILD DSS (based on the needs classification scheme) is a significant contribution towards decision-making and adds more value to traditional decision-making approaches.
- The datasets attached with modular subroutines of the integrated model are available to execute the groups of subroutines, independently, to validate the concept of loosely coupled independent subroutines.
- The MILD is a multi-layered architecture and its knowledge base is a collection of diverse independent modular subroutines. The user interface is crafted in such a way that the analyst or DSS developer has the impression that they are interacting with a single, coherent system.
- To utilize an existing model saves on development time, but it sometimes requires additional efforts to link up existing models. Our approach also provides flexibility to replace existing subroutines with new ones when decision support needs changes.
- Had the modular subroutines been run by other approaches (for example, the simple manual approach) these integrated results might not be obtained so readily. It is highlighted that the integrated model makes it possible to obtain new insights.
- The need for an integrated model in disaster management is due to the high level of complexity of decision-making problems, their multiple scales of description, and the diversity of available independent specific models.
- The decision makers and DSS developers are becoming the heaviest user of independently developed DSS models and subroutines. The availability of an integrated model and the MILD DSS framework allows them to profit from the use of existing modular subroutines and to evaluate their appropriateness.
- The MILD DSS is designed to make it easier for DSS developers to develop an integrated model with better outcomes in terms of model adoption, peer review and utilization.
Agent Based Security Framework to Protect the Integrated Model

Lack of security awareness in developing disaster management systems results in three aspects of security problems:
1. Data Integrity: Data is not secured
2. Communications are not protected from intrusion
3. Data validation and verification techniques are not properly tested
4. Physical devises are often missing or faulty
5. Different controls are not properly defined as far as security is concerned in disaster management.

In this article we devise a framework for the security of disaster management systems in general and to protect the integrated model in particular, this approach implies that care must be taken to gain an overall proactive understanding of how information is used throughout the disaster mitigation phases (for example, preparedness, mitigation, response and recovery) and its corresponding security requirements. Such approach encompasses the objectives of ensuring the availability aspect of security; and the confidentiality and integrity aspects of data. In addressing these security objectives the architecture must explore solutions keeping the following assumptions in view:

- Security attacks are those attacks that may achieve unauthorized access, to harm the integrated model.
- The framework is confined to provide effective and appropriate responses rather than only detecting attacks.
- Attacks may be internal or external.
- Attack is after it comes from a firewall and acts as a second layer in the protection hierarchy.

The framework is consists of two main components:
1. A detection agent which aims at detecting the attack
2. Solver agent which exchange the information with other agents and suggest an appropriate response by making use of a case-base of security integrity frameworks.
Based on the above assumptions we elaborate on the functional architecture of the proposed security framework designed to protect the integrated model. Security in the development of integrated model for decision support in disaster management systems can not be an afterthought. It must be integrated with the core functionality and not considered to be implemented at the end as an optional service. The proposed security framework for the protection of the integrated model based on agent technology is shown in Figure 2. The following are the key characteristics of the security framework:

- It includes autonomy, adaptability, and efficiency achieved through agents technology
- The framework is designed as simple as possible to understand and administer
- The proposed security framework is designed not only for the integrated model generated by the MILD DSS but it can easily be extended to other DSS systems for disaster management
- It provides flexibility and adaptability according to the changing needs of the disaster management domain. Such flexibility is achieved by the use of intelligent agents
- Access Control aims at preventing an agent from accessing unauthorized resources (Karnouskos, Busse et al. 1999). In our security framework calls to resources are also intercepted to promote access control.
It is therefore necessary to identify the purpose of this proposed framework. The significant role of this framework is a two-fold activity: 1) the security framework is to be deployed to protect the integrated model for decision support in disaster management by detecting attacks; and 2) when attacks are detected, the security framework deals with these attacks in a real-time environment by suggesting an appropriate security integrity framework. These two different activities are carried out by using intelligent agent technology.

Before we elaborate on the proposed framework any further it is important to highlight the meaning of “attack” on an integrated mode. According to Boudaoud et al. (Boudaoud, Labiod et al. 2000) an attack can be defined as any non-standard activity which:

- breaks privacy rules, compromising the information confidentiality;
- alters information, compromising the data integrity;
- makes a network infrastructure unavailable or unreliable, compromising the availability of a resource. In this case, we speak about denial of service attacks.

The various attacks can be classified in two categories (Boudaoud, Labiod et al. 2000):

- **External attacks:** they are generated from the outside by a hacker who is trying to access to a network to have information, have fun, or trying any kind of denial of service attack.
- **Internal attacks:** they are generated by internal users. These users abuse of their rights and privileges to do unauthorized activities and to obtain unauthorized access.

In this paper our focus is on the above-mentioned two categories of attack which are external and internal.

The base-line of the framework includes a firewall which should be capable of identifying the attacks on the first place if unsuccessful then directing this information to the detecting agent, which is handled subsequently. The proposed security framework consists of the following agents: Detection Agent, Coordination and Supervisor Agent, Network Management Agent, Solver Agent, and Security Programming and Framework Agent. Actually, these agents are mainly responsible for the detection of the attack and subsequently, proving a secured integrity framework. These agents are implemented in JAVA.

- **Network Management Agent:** Network Management Agent provides network related information and monitoring all the traffic that goes through the network.
- **Detection Agent:** The Detection Agent analyzes the observed network activities retrieved from the firewall and sends signals when it detects misuse and/or anomaly.
- **Communication and Supervisor Agent:** In this framework, KQLM has been used as agent communication language. Agents’ communication with each other is controlled and monitored by this agent. In addition it performs the job of a supervisor and establishes a connection between the communicating agents.
• **Solver Agent:** The purpose of the solver agent is to provide a security integrity framework in case of attack. This agent is assisted by the security programming and framework agent.

• **Case-Base:** Another important component of the framework is case-base which stores the frameworks of security integrity. In case of detected attack the solver agent first searches through the case-base to select the suitable security integrity framework. If the search is unsuccessful, consequently, a new framework is generated by making use of programming and framework agent.

The two important agents (detection and solver) form the foundation of the architecture. These agents are constructed as a collection of Java objects and are equipped with the identifier of the tag to which it is attached. Every agent program must be an instance of a subclass of the abstract class DetectAgent and SolveAgent respectively which are given as follows:

```java
public abstract class SecurityAgent {
    public void detect(String data) {
        for(i = 0; i < length of data; i++) {
            if (data[i] == 'ATTACK') {
                // SecurityAgent.solveAgent(data)
                next();
                // SecurityAgent.applyFramework(data),
                next();
            }
        }
    }
}

public class DetectAgent extends SecurityAgent {
    public void detect(String data) {
        if (CheckCase(data, current) == TRUE) {
            return Solution;
        } else {
            Solve(data, current);
            updateCaseBase();
            // DetectAgent.solveAgent(data, current)
        }
    }
}

public class SolveAgent extends SecurityAgent {
    public void detect(String data) {
        // SolveAgent.solveAgent(data)
    }
}
```
An Application Scenario

As previously mentioned the definition of an attack, having defined the attack, we now consider that MILD DSS is in operation and is capable of generating an integrated model for decision support in disaster management. Once the attack has occurred the security framework comes into action and protect the integrated model. So, we do not make a distinction here between an external or internal attack. Consider it as an attack, derived from either the security policies specified at a high level by the administrator or by third party. However our goal is to protect the integrated model and suggest an appropriate security integrity framework.

Having defined these notions, we now illustrate how the multi-agents system detects intrusions and how the matching of security integrity framework is performed by making use of case-base. For this, we describe briefly an example for detecting and managing attack. To detect and protect this attack, we propose the following scenario (See Figure 3): We have a distributed network constituted of a web server, hosted in a virtual network and various client systems attached to this web server. The MILD DSS is running separately and generating integrated model under the protection of the proposed agent-based security framework.

Therefore, to detect an attack, detection agent receives a detection goal containing the attack and the actions to execute the detection algorithms. If the attack is not detected than a message is passed to the interface application layer that integrated model is protected. On the other hand, if the attack is detected then it is passed to the solver agent through the communication and supervisor agent (the role of this agent is to monitor the communication between agents).

Figure 3: Example Scenario for the Protection of Integrated Model

The solver agent searches through the case-base to find the pre-defined security framework in order to protect the integrated model. If it is unsuccessful, then security
programming and framework agent generates an integrated security framework. It must be noted that attack on the MILD DSS server could occur from inside (web server) or outside. A simplified flow chat of how this framework works is shown in Figure 4.

An application scenario is introduced here in order to show the flexibility/advantages of the proposed security framework offers. We introduce the concept of agent-based security for the integrated model which facilitates effective and efficient decision-making in the area of disaster management. In any case this kind of scenario is supported by the security framework we have presented here.

![Figure 4: Simplified Flow-Chart of the proposed Framework](image)

**Conclusions and Future Research Directions**

This paper presented the overview of the conceptual architecture of the MILD-DSS (necessary to understand the integrated model). The architecture is an innovative method for developing a dynamic integrated model for decision support in disaster management. It is composed of three main layers, independent, dependent and interface. Unlike the traditionally developed DMDSS, which is based on a specific decision support need, it emphasizes the development of a dynamic integrated model based on the changing decision support needs which arises various disaster instances. The integrated model provides and facilitates effective decision-making in disaster management based on a dynamic scenario. Based on the importance of this integrated model the security issues of this model become a serious concern. In order to protect and secure the integrated model from an internal or external attack, we propose an agent-based security framework to protect the integrated model. The framework consists of two main components: (1) a detection agent which aims at detecting the attack; and (2) solver agent which exchange
the information with other agents and suggest an appropriate response by making use of a case-base of security integrity frameworks. An application scenario has been introduced to highlight the usefulness and effectiveness of the proposed framework. This has been shown that the proposed security framework is capable to protect the integrated model from internal as well as external attacks. In future, we will report the extension of this security model such that its scope becomes wider and generalized to other integrated models developed for decision-making in disaster management.

References


