A Security Decision-Reaction Architecture for Heterogeneous Distributed Network

Feltus, Christophe; Khadraoui, Djamel; Aubert, Jocelyn

Published in:
Proceedings of the The Fifth International Conference on Availability, Reliability and Security ("ARES 2010 - The International Dependability Conference"), Krakow, Poland

DOI:
10.1109/ARES.2010.57

Publication date:
2010

Document Version
Early version, also known as pre-print

A Security Decision-Reaction Architecture for Heterogeneous Distributed Network

Christophe Feltus
Public Research Center Henri Tudor
Luxembourg-Kirchberg, Luxembourg
PReCISE Research Centre, Faculty of Computer Science,
University of Namur, Belgium
christophe.feltus@tudor.lu

Djamel Khadraoui, Jocelyn Aubert
Centre for IT Innovation
Public Research Centre Henri Tudor
29, Avenue John F. Kennedy, L-1855 Luxemburg
djamel.khadraoui@tudor.lu, jocelyn.aubert@tudor.lu

***Abstract*— The main objective of this paper is to provide a global decision-reaction architectural built on the requirements for a reaction after alert detection mechanisms in the frame of information systems security and more particularly applied to telecom infrastructures security. These infrastructures are distributed in nature, therefore the architecture is elaborated using the multi-agents system that provides the advantages of autonomous and interaction facilities, and has been associated to the ontoBayes model for decision support mechanism. This model helps agents to make decisions according to preference values and is built upon ontology based knowledge sharing, bayesian networks based uncertainty management and influence diagram based decision support. The Multi-Agent System decision-reaction architecture is developed in a distributed perspective and is composed of three basic layers: low level, intermediate level and high level. The proposed approach has been illustrated based on the network architecture for heterogeneous mobile computing developed by the BARWAN project. Accordingly: the Building Area constitutes the low level and aims to be the interface between the main architecture and the targeted infrastructure. The Campus-Area is the intermediate level responsible of correlating the alerts coming from different domains of the infrastructure and to smartly deploy the reaction actions.**

Keywords-security; decision system; recaction; distributed network; bayesian network; multi agent system.

I. INTRODUCTION

Today information systems and mobile computing networks are more widely spread and mainly heterogeneous. This basically involves more complexity through their opening, their interconnection, and their ability to make decisions [1]. Consequently, this has a dramatic drawback regarding threats that could occur on such networks via dangerous attacks [2]. This continuously growing amount of carry out malicious acts encompasses new and always more sophisticated attack techniques, which are actually exposing operators as well as the end user.

State of the art in terms of security reaction is limited to products that detect attacks and correlate them with a vulnerability database but none of these products are built to ensure a proper reaction to attacks in order to avoid their propagation and/or to help an administrator deploy the appropriate reactions [3, 4]. In the same way, [5] says that at the individual host-level, intrusion response often includes security policy reconfiguration to reduce the risk of further penetrations but doesn't propose another solution in term of automatic response and reaction. It is the case of CISCO based IDS material providing mechanisms to select and implement reaction decision.

Information security management and communication systems is actually in front of many challenges [6] due to the fact that it is very often difficult to establish central or local permanent decision capabilities, have the necessary level of information, quickly collect the information, which is critical in case of an attack on a critical system node, or launch automated counter measures to quickly block a detected attack.

Based on that statements, it appears crucial to elaborate a strategy of reaction after detection against these attacks Our previous work around that topic has provided first issues regarding that finding and has been somewhat presented in [6] and [7]. These papers have proposed an architecture to highlight the concepts aiming at fulfilling the mission of optimizing security and protection of communication and information systems which purpose was to achieve the following:

- Reacting quickly and efficiently to any simple attack but also to any complex and distributed ones;
- Ensuring homogeneous and smart communication system configuration, that are commonly considered and the main sources of vulnerabilities.

One of the main aspects in the reaction strategy consists of automating and adapting policies when an attack occurs. In scientific literature a large number of definitions for policy and conceptual model exist. The most famous are Ponder [8] and Ponder2 [34], Policy Description Language [9] and Security Policy Language [10]. For the purpose of that paper, we prefer the one provided by Damianou et al. in [8]: *Policies are rules that govern the behavior of a system.*

The provided policy adaptation is considered as a regulation process. The main steps of the policy regulation are described in Fig. 1, which shows the process that takes the business rules as input, and maps them onto technical policies. These technical policies are deployed and instantiated on the infrastructure in order to have a new state of temporary network security stability adapted to the ongoing attack. This policy regulation is thereafter achieved in modifying/adding new policy rules to reach a new

standing (at least up to the next network disruption) policy based on the observation of the system's current situation.

In this paper, we focus our work on policy deployment and on policy modification decision-reaction challenges as highlighted in the rounded rectangle of Fig. 1. This twofold challenge has already been addressed by other researches like in [11]. Torrellas explains that facilitating timely decision-making may achieve much greater productivity benefits by engineering network security systems using multi-agents. In [12], Yu developed the concepts of tele-service and proposed an implementation of an e-maintenance platform based on a Mulit Agent System (MAS). Yu explained how a Case-Based Reasoning [13] method may be used to improve the autonomous decision-making ability. Others' works propose rather similar solutions like [14, 15] but none are explicitly dedicated to the management of security alerts reaction in the field of open and heterogeneous networks. Consequently, the combination of the reaction mechanism with the decision support system remains, for those solutions, a poorly addressed requirement in parallel to other more specific constraints related to the characteristics of the context.

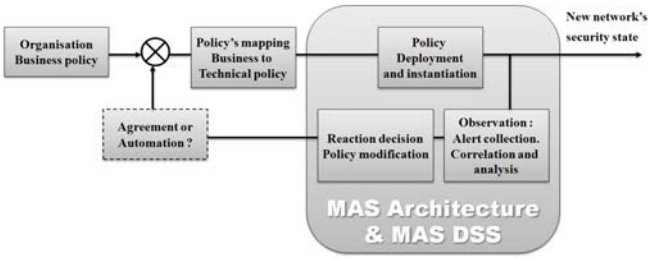

Figure 1. Policy regulation

To illustrate this decision mechanism, we use the results of the BARWAN[1] project. This project focused on enabling *truly useful mobile networking across an extremely wide variety of real-world networks and mobile devices*. The case study analyzed by the project is a medical application enabled by wide-area wireless and that exploits the Berkeley InfoPad[35] pooled computing power to permit a small number of workstations to support a large number of end users. Fig.3 highlights the distribution of the application over the buildings, the campus and the metropolitan layers. In that paper, an architecture is proposed to adapt a reaction once an attack occur on one of those layers. Additionally, the architecture makes is possible to integrate internal or external contextual information for the reaction decision like, i.e. the usage of the application, as proposed in the case study, during a medical rescue operation after a serious auto accident on Golden Gate Bridge[2].

[1] Bay Area Research Wireless Access Network project, conducted at the University of California at Berkeley.

[2] The complete case study is available on http://bnrg.eecs.berkeley.edu/~randy/Daedalus/BARWAN/BARWAN_application.html

The next section introduces the MAS architecture, section 3 exposes the decision support system as well as its combination with the MAS, and the last section concludes the paper.

II. MULTI AGENT SYSTEM ARCHITECTURE

MAS is composed of several agents, capable of mutual interaction. The interaction can be in the form of message passing or producing changes in their common environment. Agents are pro-actively, reactively and socially autonomous entities able to exhibit organized activity, in order to meet their design objectives, by eventually interacting with users. An agent is collaborative by being able to commit itself to society or/and another agent.

An agent encapsulates a state and a behavior and provides moreover a number of facilities such as: control of its behavior, the ability to decide even if external events influence its decision, the possibility to exert its control in various manners (reactively, directed by goals, socially). Moreover, MAS have several control flows while a system with objects has a priori only one control flow.

The agents also have global behavior within the MAS, such as the cooperation (agents share the same goal), collaboration (agents share intermittently the same goal) or competition (incompatible goals between agents).

To manage several different systems, due to their location, their business domain or their organization type, a distributed system is appropriate. Furthermore, a distributed solution brings some autonomy to the managed systems. Robustness, survivability and availability are also impacted.

The distributed architecture introduced in this paper is composed by several components, called "operators", which have different responsibilities. Those operators are organized in two dimensions, as presented in Fig. 2.

The vertical dimension, structured in layers relative to the managed network organization, allows adding abstraction in going upward. Indeed, the lowest layer is closed to the managed system and thus being the interface between the targeted network and the management system. The higher layer exposes a global view of the whole system and is able to take some decisions based on a more complete knowledge of the system, business, and organization. Intermediate levels (1 to n-1) guarantee flexibility and scalability to the architecture in order to consider management constraints of the targeted infrastructure. Those middleware levels are optional but allow the system to be better adapted to the complexity of a given organization and the size of the information system.

The horizontal dimension, containing three basic components, is presented in Fig. 2 and its three main phases are described below:

1) Alert: Collect, normalize, correlate, analyze the alerts coming from the managed networks and represent an intrusion or an attack. If the alert is confirmed and coherent, it is forwarded to the reaction decision component. (Alert Correlation Engine-ACE).

2) Reaction Decision: Receive confirmed alerts for which a reaction is expected. Considering the knowledge of:

policy, the systems' organization and specified behavior, these components decide if a reaction is needed or not and define the reaction, if there is any. The reaction will be modification(s), addition(s) or removal(s) of current policy rules. (Police Instantiation Engine-PIE).

3) Reaction: Instantiation and deployment of the new policies, on the targeted networks. The deployment (Policy Deployment Point – PDP) and enforcement (Policy Enforcement Point – PEP) of these new policies, lead to a new security state of the network. The terminology in italic used in section 4 is extracted from both: XACML [16] and OrBAC Model [17, 18].

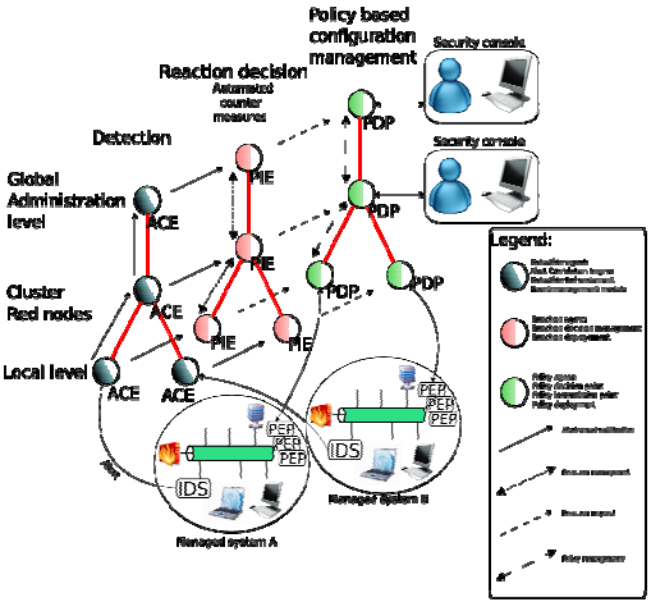

Figure 2. Reaction Architecture Overview

An issue is raised considering which layer is allowed to take a decision reaction: only one layer, two, several, or all? If more than one layer can trigger a reaction on the same object(s), there will be a conflict issue. Thus, the system should be able to provide mechanisms to solve conflicts between several selected reactions. Another issue concerns the agreement: at which level should it be asked? A solution could be to ask at the same level (or at an upper one) that the reaction decision is made; this should be specified by the user. A possible solution is a distributed, vertically layered and hierarchical architecture. The layer's number could be adapted according to the managed systems' organization. In our case, three layers are sufficient (local, intermediate and global). The reaction system is composed of three main parts: the alert management part, the reaction part and the police definition-deployment part. Three trees (alert, reaction and policy) could be placed side by side, as presented in Fig. 2. These trees are alike but their operators have different functions. The alert tree collects the alerts with the local operators and correlate them in several steps, one step by layer.

Fig 3. explains how the reaction architecture is mapped onto the BARWAN network (borrowed from [33]). The three layers are from top to bottom: The metropolitan Area, The campus area, and the in-building network (building A and B).

The next step of our research development is firstly the definition of a reaction engine that encompasses both, architecture components and the communication engine between these components. This engine is based on a message format and on a message exchange protocol based on standards such as [19]. Secondly, real cases are studied in order to experiment with the architecture and its associated protocol.

The message format is defined in XML format and is structured around a number of attributes that specify the message source, the message destination and the message type (alert, reaction, policy request, policy modification, policy modification validation, decision and synchronization). The protocol defines the exchange format and the workflow of messages between the architecture components. It encompasses a set a rules governing the syntax, semantics, and synchronization of communication. The technical requirements request the operator structure must be flexible in order to be able to reorganize itself, if an operator fails or disappears. Each operator also has to be autonomous in order to permit reorganization. Given these requirements, the use of a MAS appears as a solution to provide autonomy, flexibility and decision mechanisms to each operator that are consequently represented by agents.

As studied in the state of the art presented in [20], a set of agents could be managed and controlled through an organization. An organization is a set of agents playing roles, gathered in a normative structure and expecting to achieve some global and local objectives. Several models like the roles model, the tasks model, the interaction model or the norms models specify an organization.

In our context we need an interaction definition in order to specify communication protocols between agents representing operators. We also need roles in order to specify which agent will have to communicate or act in order to detect intrusions and then react. Based on this needs, the use of an electronic institution based on agents is one of the possibilities that we will investigate.

The main goal of the reaction policy enforcement engine is to apply policies in terms of specific concrete rules on "technical" devices (firewall, fileserver, and other systems named PEP). For that, we need means to make ACE, PIE, PDP and PEP interact and collaborate.

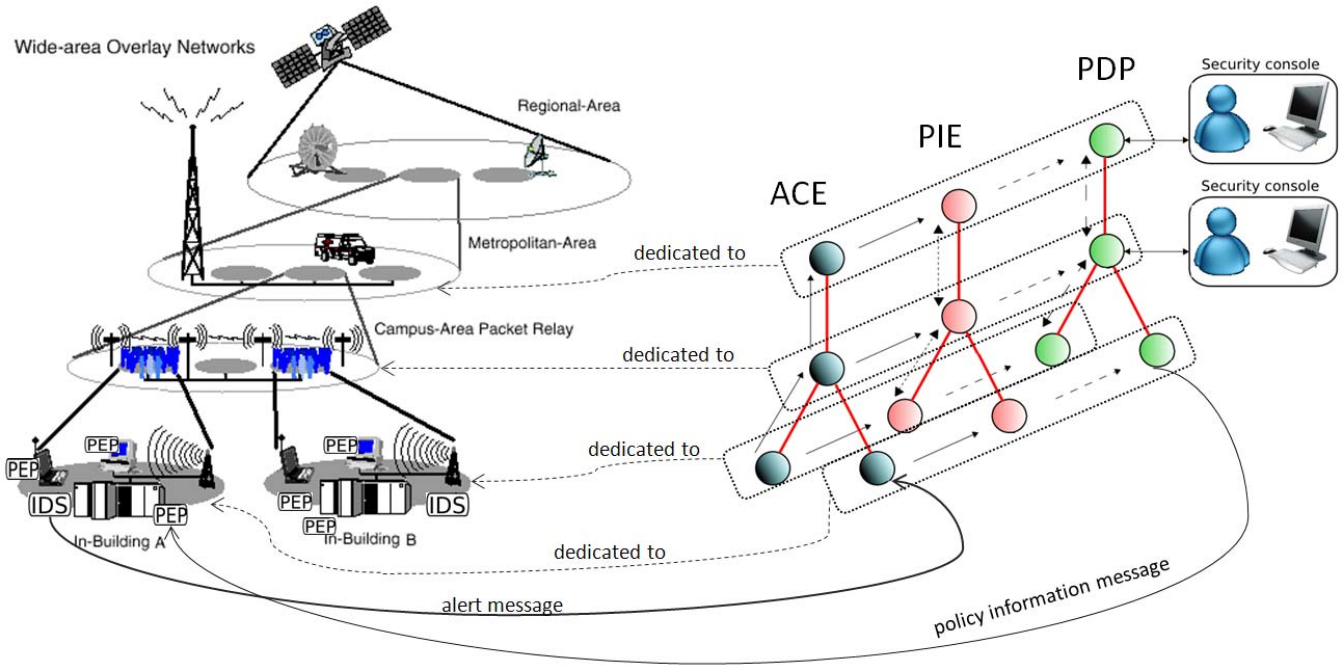

Figure 3. Mapping of the BARWAN architecture with the Multi-Agent System reaction architecture

The multi-agents systems concept already defines architectures and models for autonomous agents' organization and interaction. Existing platforms like JADE (Java Agent DEvelopment Framework) [21, 22] implement agents' concepts as well as their ability to communicate by exchanging messages and the reaction components integration could be simplified. This is a solution, which will be detailed hereafter. The Foundation for Intelligent Physical Agents (FIPA) [23] promotes the success of emerging agent-based applications, services and equipment. It makes available internationally agreed specifications that maximize interoperability across agent based applications, services and equipment pursue this goal. This is realized through open international collaboration of member organizations, which are companies and universities active in the agent field. FIPA's specifications are publicly available. They are not technologies for specific application, but generic technologies for different application areas, and not just independent technologies but a set of basic technologies that can be integrated by developers to make complex systems with a high degree of interoperability.

The used multi-agent framework is JADE. We base ourselves on a survey made in [24] to argue that this agent platform responds to the expectations in terms of agents' functionalities, security, performance, standardization, and secure communication between agents.

Fig. 4 introduces the developed architecture. The flow is supposed to begin with an alert detected by the IDS positioned on the InfoPad server. This alert is send to the BuildingA_ ACE agent. This ACE agent confirms or not the alert to the PIE. This decision to confirm the alert is explained in section 3. Afterwards, the PIE decides to apply new policies or to forward the alert to an ACE from a higher layer (upper ACE). Its PIE agent sends the policies to the PDP agent, which decides which PEP is able to implement it in terms of rules or script on devices (InfoPad server, fileserver, etc.) Then, the PDP agent sends the new policy to the InfoPad PEP agent that knows how to transform a policy into a rule or script understandable by the InfoPad server.

On Fig. 4, dash dot lines stand for flow of messages encompassing alert or alert confirmation. Full lines stand for flow of messages containing policies information, and dot lines are reserved for decision support mechanisms.

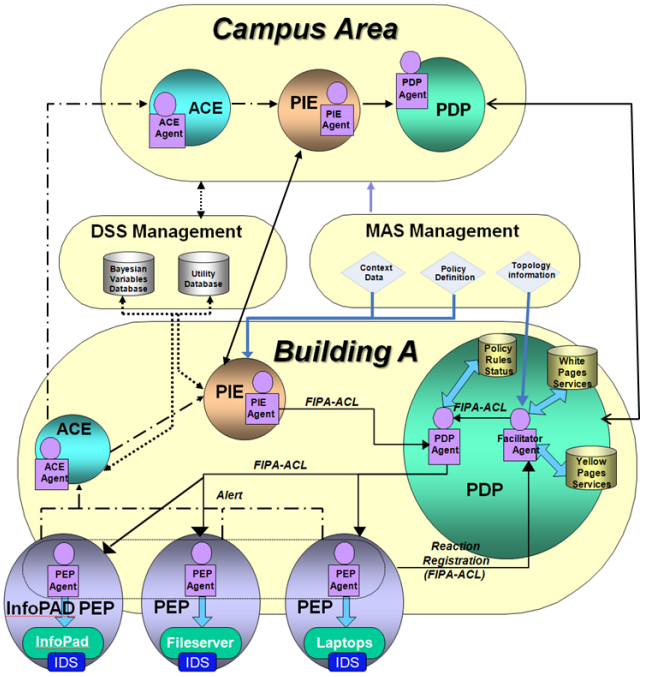

Figure 4. Multi-Agent System reaction architecture

A focused analysis of the PDP shows that it is composed by several modules. For the multi-agent system point of view, the Component Configuration Mapper results from the interaction between the PDP agent and the Facilitator Agent while the Policy Analysis module is realized by the PDP agent. The Facilitator manages the network topology by retrieving PEP agents according to their localization (devices registered with IP address or MAC address) or according to actions they could apply and their type (firewall, file server, etc.). For that the Facilitator uses white pages and yellow pages services. The JADE platform already provides implemented facilitator and searching services. Besides, the use of a multi-agent system as the framework provides flexibility, openness and heterogeneity. Actually, when we decide to add a new PEP, we just have to provide its PEP Agent with the ability to concretely apply the policies that will register itself through the Facilitator, which will update the databases.

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

III. DECISION SUPPORT ARCHITECTURE

Section 2 explains the developed MAS architecture that guarantees a telecommunication security incident reaction. Section 3 explains the implementation of the decision mechanism. The MAS architecture has voluntarily been explained before the Decision Support System (DSS) part because components of this architecture are used for the illustration of the DSS.

One important challenge of the DSS is the management of uncertainty. Uncertainty is defined as situation "*caused by a lack of knowledge about the environment when a gents need to decide the truth of statement.*"

Decision is a process [25] and consequently, it may be represented using its input and its output. For the security incident reaction, inputs of the alert sending decision mechanism are for instance: the severity, duration and frequency of the alerts, the contribution of the system to the medical rescue operation (if any), or the criticality of that rescue operation. Outputs of the process are for instance: the escalation of the alert to upper ACE or its confirmation to the PIE. For the clarity of the paper, some parameters from the case study will be partially omitted.

As explained by Yang [26], the decision-making mechanism is composed of four pillars: Ontology, Bayesian Networks (BN), Influence Diagram (ID) and Virtual Knowledge Community (VKC). In the framework of that paper, the VKC will not be treated because the use of the 3 first pillars is enough to understand the decision mechanism. The approach preferred to design the decision mechanism is adapted from the research performed by Yang's thesis for the incident reaction through a MAS architecture. As a consequence our solution differs from and completes the Yang research since our DSS is illustrated by a real architecture for incident reaction that is really deployed in our research labs.

A. Ontology

Ontology is the first pillar and is defined by *a formal, explicit specification of a shared conceptualization* [27]. Ontology may be categorized as domain ontology when it concerns concepts and their relations from a same and well-defined domain or top-level ontology when it concerns very general domain-independent concepts. Ontology is the most import pillar in that, it will be adapted to support the second pillar concerning the Bayesian Network and the third pillar concerning the Influence Diagram.

For the incident reaction system, ontology is defined using the Web Ontology Language (OWL). Resource Development Frameworks (RDF) syntax is the most commonly used method to model information or meta-concepts in OWL. It may be implemented in web resources and is structured based on the triple (object, subject, predicate). Fig. 5 illustrates RDF graph. Both, object and subject are resources whereas predicate is an attribute or a relation used to describe a resource.

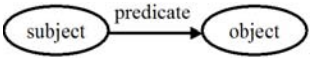

Figure 5. RDF graph

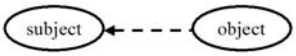

Figure 6. Dependency graph

In parallel to the MAS architecture developed in section 3, we need a DSS to decide the transfer of an alert from the IDS to the BuildingA ACE[3], for the forward of that alert to an upper ACE, and for the confirmation of the alert to the PIE. This is formalized using OWL as explained in Fig. 7. On that figure, ovals stand for OWL class, solid arrow lines stand for RDF predicate, dash arrows for influence relations and rounded rectangles for set of domain value.

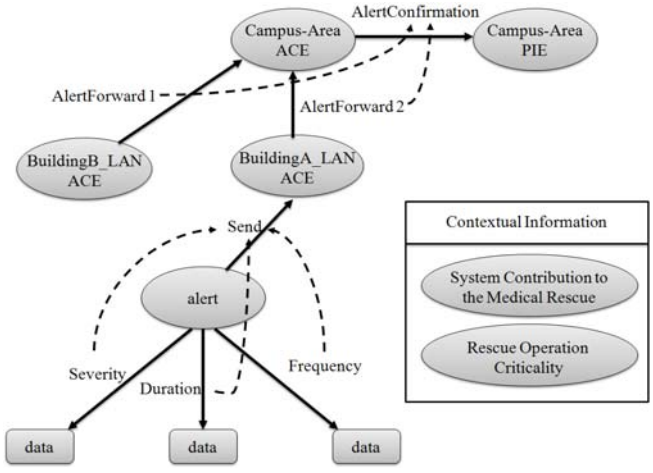

Figure 7. Decision system for alert transfer using OWL

[3] ACE agent in the BuildingA Local Area Network where alert is sent

B. *OntoBayes*

Ontology developed in the previous section permits to formalize the concept encompassed in the MAS architecture as well as their relations. However, at that the ontological level of formalization, uncertainty challenge remains unaddressed and decision mechanism remained needed for the agents to take the decision.

OntoBayes is an extension of OWL with two features: Bayesian Networks and Influence Diagram. BN address the uncertainty and ID support the decision mechanism process.

1) Bayesian networks extension

In probabilistic, Bayes Theorem is a simple mathematical formula used for calculating conditional probabilities [28]. It means that the calculations of probability depend on prior knowledge that could be considered as uncertain. I.e.: the probability of having a high impact on the medical rescue if we have before an alert of medium severity. This probability is written P(alert.severity|rescue.impact).

The BNs extension of OWL introduces the parameters of that formula by specifies the following two perspectives: a qualitative perspective and a quantitative perspective. The qualitative perspective specifies the random variables explicitly as well as their dependencies and the later associates' quantitative information to those variables.

The specification of random variable and their dependency is performed by introducing the new OWL property element `<owl:ObjectProperty rdf.ID="dependsOn"/>` and could be graphical represented as illustrated on Fig. 6.

Accordingly, the qualitative extension may be represented by 2 Bayesian graph models (Fig. 8) extracted from the OWL graph model from Fig. 6.

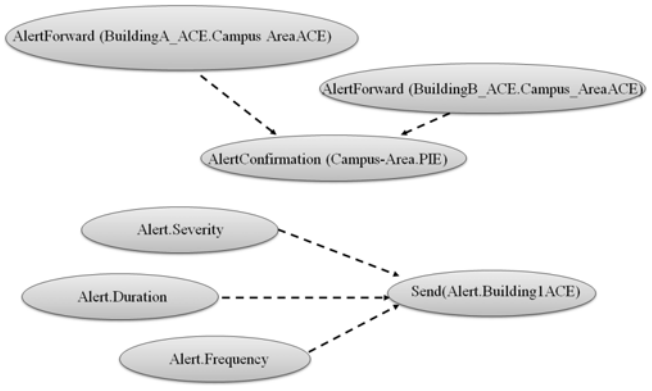

Figure 8. Bayesian graph models for alert sending and alert confirmation processes

The ovals represent Bayesian variables and the arrows specify their relations. The graph is to be read i.e. 1.: The alert that is forwarded from the BuildingB ACE to the network upper ACE has influence on the confirmation of the alert that is send from the Campus-Area ACE to the PIE. I.e. 2.: The severity of the alert has influence on the action to send an alert to the BuildingA ACE. The last examples may be translated using the new OWL `dependsOn` element as following :

```
<owl:Class rdf:ID="alert.severity">,
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty red:ID="dependsOn"/>
      </owl:onProperty>
    <owl:hasValue rdf:resource="system.impact">
  </owl:Restriction>
</owl:class>
```

Figure 9. Dependency encoding

The quantitative extension is performed with the association of probability table to the Bayesian variables. In the case of the BARWAN example, the Table 1 provides de quantitative probability P(alert.severity|rescue.impact) and is represented on Fig. 4 by the Bayesian variables database.

TABLE I. BAYESIAN VARIABLES VALUE PROBABILITY

ProbCell.	HasPParameters	HasPValue
Cell_1	alert.severity=low|rescue.impact=low	0,8
Cell_2	alert.severity=medium|rescue.impact=low	0,4
Cell_3	alert.severity=high|rescue.impact=low	0,1
Cell_4	alert.severity=low|rescue.impact=medium	0,3
Cell_5	alert.severity=medium|rescue.impact=medium	0,9
Cell_6	alert.severity=high|rescue.impact=medium	0,5
Cell_7	alert.severity=low|rescue.impact=high	0,1
Cell_8	alert.severity=medium|rescue.impact=high	0,4
Cell_9	alert.severity=high|rescue.impact=high	0,7

The conditional probability from Table 1 is encoded as follows (Fig. 10):

```
<owl:Class rdf:ID="Alert">
  <CondProbDist rdf:ID="table_1">
    <hasPCell>
      <ProbC rdf:ID="Cell_1">
        <HasPValue rdf:Iddatatype="#float">
        >0,8</HasPValue>
        <HasParameters rdf:datatype="#string"
        >alert.severity=low|rescue.impact=low>
        </HasParameters>
      </ProbC>
    </HasPCell>
    ...
  </CondProbDist>
</owl:Class>
```

Figure 10. Bayesian variables value probability encoding

2) Influence diagrams extension

IDs extension aims at representing and analyzing a decisional model to support the decision-making process. The review of the literature that treats ID [29,30] shows that decision mechanisms are composed by three types of nodes: 1) Chance nodes that represent variables that are not controled by the decision maker, 2) Decision nodes that represent choices available for the decision maker, and 3) Utility nodes that represent agent utility functions. Additionally, [31] explains that three type of arcs express the relationship between nodes: I) Information arcs (`isKnownBy`) that point out the information that is necessary for the decision maker, II) Conditional arcs (`influenceOn`) that point out the probabilistic dependency on the associated variable, and III) Functional arcs (`attributeOf`) that point out variables used by utility nodes as decision criteria.

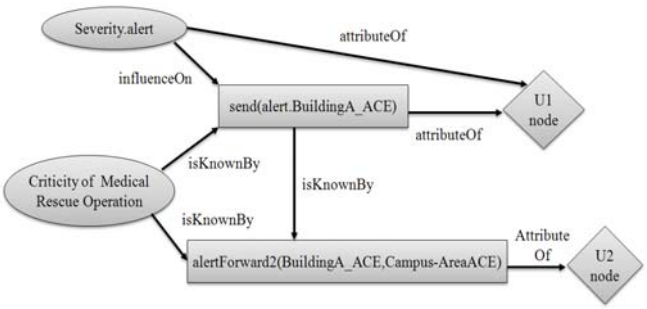

Figure 11. ID's graph model of alert transfer

Based on that structure of decisional model, the alert transfer may be represented in Fig. 11. Ovals stand for Chance nodes, rectangles stand for Decision nodes, and diamonds stand for Utility nodes. The information arc relates to all information observed to make a decision and the conditional arc relates to data issued from Chance node and considered as evidence for the Decision nodes.

Additionally, to make a decision, the agent that takes a decision needs to have its preferences quantified according to a set of attributes. In [32], Butler introduces the theory of multi-attribute utility (MAUT) to quantify a preference with numerical value. The most important preference has the higher value whereas the worst has the lower one. To achieve that, the Utility node is associated with a utility table that gathers the preferences of all decision choices. Table 2 shows these preferences for the BuildingA ACE alert sending decision taking mechanism and is represented by the utility database in Fig. 4.

TABLE II. UTILITY TABLE FOR IN-LAN ACE ALERT SENDING

UtilityCell.	HasUParameters	hasUValue
Cell_1	send(alert.BuildingA_ACE)=yes|severity.alert=low	-80
Cell_2	send(alert.BuildingA_ACE)=yes|severity.alert=medium	50
Cell_3	send(alert.BuildingA_ACE)=yes|severity.alert=high	100
Cell_4	send(alert.BuildingA_ACE)=no|severity.alert=low	80
Cell_5	send(alert.BuildingA_ACE)=no|severity.alert=medium	40
Cell_6	send(alert.BuildingA_ACE)=no|severity.alert=high	-100

The Fig. 13 shows the encoding of Table 2 utility table for BuildingA_ACE alert sending :

```
owl:Class rdf:ID="send(alert.BuildingA_ACE)">
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="attributeOf"/>
      </owl:onProperty>
    <owl:hasValue rdf:resource=#U
  </owl:Restriction>
     ...
  <rdfs:subClassOf>
    <owl:hasValue rdf:ID="DecisionNode"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="severity.alert">
     ...
  <rdfs:subClassOf>
    <owl:hasValue rdf:ID="ChanceNode"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="U1">
  <UtilityTable rdf:ID="table_1">
    <hasUCell>
      <UtilityCell rdf:ID="cell_1">
        <hasUParameter rdf:datatype="#string"
>send(alert.BuildingA_ACE)=yes,severity.alert=low
        </hasUParameter>
        <hasUValue rdf:datatype="#float"
        >-80</hasValue>
      </UtilityCell>
    </hasUCell>
     ...
    <hasUCell>
      <UtilityCell rdf:ID="cell_6">
        <hasUParameter rdf:datatype="#string"
        > send(alert.BuildingA_ACE)=no,
        > severity.alert=high
        </hasUParameter>
        <hasUValue rdf:datatype="#float"
        >-100</hasValue>
      </UtilityCell>
    </hasUCell>
..</UtilityTable>
</owl:Class>
```

Figure 12. Utility encoding

As seen in Fig. 6, a sequential path between all decisions exists. Indeed, some decision depends on previous decisions and as a consequence, previous decisions (decision node) become chance nodes for next chance node. Fig. 11 illustrates that send(alert.BuildingA_ACE) is at the same time a decision node and a Chance node that is known be the decision node alertForward2(BuildingA_ACE,Campus-AreaACE).

IV. CONCLUSIONS

In this paper we have presented a global and integrated decision-reaction architecture developed for an incident reaction system and based on a policy regulation approach strategy. The solution is composed firstly with a MAS that offers the advantage to react quickly and efficiently against an attack while being adapted for heterogeneous and distributed networks and secondly with a decision support system that helps agents to make decisions based on utility preference values. This is achieved by taking uncertainty into account through Bayesian networks and influence diagram.

The architecture has been illustrated based on the network architecture for heterogeneous mobile computing developed by the BARWAN project. Accordingly, contextual information has been introduced in the decision mechanism like i.e. the criticality of the medical rescue operations.

The decision support system has been explained for the transfer of an alert from the alert correlation engine to the policy instantiation engine. Other decision points exist in the architecture. All of them could be solved using decision support system but they are not explained in the paper.

The future works based on our achievements will be the specification of a protocol, specification of the messages and thus the reaction methodology service oriented based. This protocol and methodology will be dedicated to the architecture presented in this paper and address the interoperability issues with regard to the policy representation and modeling.

ACKNOWLEDGMENT

This research was funded by the National Research Fund of Luxemburg in the context of TITAN (Trust-Assurance for

Critical Infrastructures in Multi-Agents Environments, FNR CO/08/IS/21) project.

REFERENCES

[1] A. Cuevas, P. Serrano, J. I. Moreno, C. J. Bernardos, J. Jähnert, R. L. Aguiar, V. Marques, Usability and Evaluation of a Deployed 4G Network Prototype, *Journal of Communications and Networks*, Vol. 7 (2), 2008.

[2] Teo, Joseph Chee Ming; Tan, Chik How; Ng, Jim Mee, Denial-of-service attack resilience dynamic group key agreement for heterogeneous networks, *Telecommun. Syst*. 35, No. 3-4, 141-160 (2007).

[3] L. J. LaPadula. State of the Art in Anomaly Detection and Reaction Technical Report MP 99B0000020, Mitre, July 1999.

[4] G.L.F. Santos, Z. Abdelouahab, R.A. Dias, C.F.L. Lima, E. Nascimento , E.M. Cochra. An Automated Response Approach for Intrusion Detection Security Enhancement, *Software Engineering and Applications*, 2003.

[5] M. Petkac and L. Badger, Security agility in response to intrusion detection in *16th Annual Conference on Computer Security Applications* (ACSAC '00), 2000.

[6] C. Feltus, D. Khadraoui, B. de Rémont and A.Rifaut, Business Gouvernance based Policy regulation for Security Incident Response. *IEEE Global Infrastructure Symposium*, 6 July 2007.

[7] Gateau, D. Khadraoui, C. Feltus, Multi-Agents System Service based Platform in Telecommunication Security Incident Reaction, *IEEE Global Information Infrastructure Symposium, 2009.*

[8] N. Damianou, N. Dulay, E. Lupu, M. Sloman , The Ponder Policy Specification Language, *Workshop on Policies for Distributed Systems and Networks* (Policy2001), HP Labs Bristol, 29-31. Springer-Verlag.

[9] Bertino, E., Mileo, A., and Provetti, A. 2005. PDL with Preferences. IEEE international *Workshop on Policies For Distributed Systems and Networks*, Policy 2005 – Vol. 00, IEEE Computer Society, Washington, DC, 213-222.

[10] Basile, C.; Lioy, A.; Perez, G. Martinez; C., F. J. Garcia; Skarmeta, A. F. Gomez, POSITIF: A Policy-Based Security Management System, *Policies for Distributed Systems and Networks*, 2007. POLICY'07, pp. 280 – 280.

[11] Torrellas, G.A.S, Modelling a network security systems using multi-agents systems engineering, IEEE International Conference on Systems, Man and Cybernetics, 2003. Vol 5, (5-8). 2003 pp 4268 - 4273.

[12] R. Yu, B. Iung, H. Panetto, A multi-agents based E-maintenance system with case-based reasoning decision support, *Engineering Applications of Artificial Intelligence*, Vol. 16, Issue 4, June 2003, Pages 321-333

[13] Aamodt, A., Plaza, E., 1994. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications IOS Press 7 (1), 39–59.

[14] K.-Y. Lu, C.-C. Sy, A real-time decision-making of maintenance using fuzzy agent, *Expert Systems with Applications*, Volume 36, Issue 2, Part 2, March 2009, Pages 2691-2698

[15] Carrascosa et al., 2006 C. Carrascosa, J. Bajo, V. Julian, J.M. Corchado and V. Botti, Hybrid multi-agent architecture as a real-time problem-solving model, Expert Systems with Applications 34 (2006), pp. 2–17.

[16] http://xml.coverpages.org/draft-seitz-netconf-xacml-00.txt

[17] Cuppens, F., Cuppens-Boulahia, N., Miège, A.: Inheritance hierarchies in the Or-BAC Model ad application in a network environment. In: *Second Foundations of Computer Security Workshop* (FCS'04), Turku, Finland (2004).

[18] F. Cuppens and A. Miège, Modelling contexts in the Or-BAC model, *19th Annual Computer Security Applications Conference*, Las Vegas, December, 2003

[19] IDMEF/RFC4765, Network Working Group: Hervé Debar, France Telecom; D. Curry, Guardian; B. Feinstein, *SecureWorks*, Inc.; March 2007

[20] B. Gâteau. Modélisation et Supervision d'Institutions Multi-Agents. Ph.D. Thesis, Ecole Supérieure des Mines de Saint-Etienne, 2007.

[21] F. Bellifemine, A. Poggi, G. Rimassa. JADE - A FIPA-compliant agent framework, CSELT internal technical report. Part of this report has been also published in *Proceedings of PAAM'99*, London, April 1999, pp.97-108

[22] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE - A White Paper. Sept. 2003

[23] FIPA, http://www.fipa.org/

[24] E. Bulut, D. Khadraoui, and B. Marquet, Multi-Agent based Security Assurance Monitoring System for Telecommunication Infrastructures, *Communication, Network, and Information Security conference* (CNIS 2007), Berkeley, California, USA, september 2007.

[25] H. D. Lasswell, The decision process; seven categories of functional analysis, College of Business and Public Administration, University of Maryland, 1956

[26] Y. Yang. A framework for decision support systems adapted to uncertain knowledge, PhD. Thesis, 2007. University of Karlsruhe.

[27] R. Studer, V. R. Benjamins, and D. Fensel, Knowledge engineering: Principles and methods, *Data & knowledge engineering*, 25(1-2):161-197, 1998

[28] http://plato.stanford.edu/entries/bayes-theorem/

[29] R. A. Howard and J.E. Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, September 2005.

[30] Finn V. Jensen. Bayesian networks and decision graphs. Springer, corr. print. edition, 2001.

[31] J.A. Tatman and R.D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):365–379, 1990.

[32] J. Butler, D. J. Morrice, and P. W. Mullarkey. A multiple attribute utility theory approach to ranking and selection. *Management Science*, 47(6):800–816, June 2001.[i]

[33] Eric A. Brewer, Randy H. Katz, Elan Amir, Hari Balakrishnan, Yatin Chawathe, Armando Fox, Steven D. Gribble, Todd Hodes, Giao Nguyen, Venkata N. Padmanabhan, Mark Stemm, Srinivasan Seshan, Tom Henderson, A network Architecture for Heterogeneous Mobile Computing, *IEEE Personal Communications Magazine*, Oct. 1998 http://citeseer.ist.psu.edu/article/brewer98network.html

[34] Brodie, C., George, D., Karat, C., Karat, J., Lobo, J., Beigi, M., Wang, X., Calo, S., Verma, D., Schaeffer-Filho, A., Lupu, E., and Sloman, M. 2008. The Coalition Policy Management Portal for Policy Authoring, Verification, and Deployment. In Proceedings of the 2008 IEEE Workshop on Policies For Distributed Systems and Networks - Volume 00 (June 02 - 04, 2008). POLICY. IEEE Computer Society, Washington, DC, 247-249.

[35] Truman, T. E., Pering, T., Doering, R., and Brodersen, R. W. 1998. The InfoPad Multimedia Terminal: A Portable Device for Wireless Information Access. IEEE Trans. Comput. 47, 10 (Oct. 1998).