# Designing Security Policies for Complex SCADA Systems Protection

Khadraoui, Djamel and Feltus, Christophe

LIST

22 April 2015

# Designing Security Policies for Complex SCADA Systems Protection

Djamel Khadraoui and Christophe Feltus
Luxembourg Institute of Science and Technology, 5 Avenue des Hauts-Fourneaux,
L-4362 Esch-sur-Alzette, Luxembourg
email: firstname.name@list.lu

*Abstract*—**The management and protection of these SCADA systems must constantly evolve towards integrated decision making and policy driven by cyber security requirements. The current research stream in this domain aims, accordingly, to foster the smartness of the field equipment which exist through the generic concept of SCADA management and operation. Those components are governed by policies which depend on the components roles, as well as on the evolution of the crisis which also confer to the latter the latitude to react based on their own perception of the crisis evolution. Their latitude is calculated based on the component smartness and is strongly determined by, and depending on, the cyber safety of the component environment. Existing work related to crisis management tends to consider that components evolve and are organized in systems but as far as we know, no systemic solution exists which integrates all of the above requirements. This paper proposes an innovative version of ArchiMate® for the SCADA components modelling purpose to enrich their collaborations and, more particularly, the description of their behavior endorsed in the cyber-policy. Our work has been illustrated in the frame of a critical infrastructure in the field of petroleum supply and storage networks.**

***Keywords- ArchiMate; metamodel; SCADA; multi-components system; trust; petroleum supply chains: critical infrastructure.***

## I. INTRODUCTION

Up to now, components represented at the business layers [1][2][7][8] have been considered human actors playing business roles. However, rising security requirements for the management of heterogeneous and distributed architectures calls for a rethinking of distribution of the security procedures in both: human and software autonomous entities. Although having been handled by human employees for years, the management of complex systems, nowadays, needs to be shared with intelligent software items, often perceived being more adapted to act in critical situations.

This statement is enforced by the characteristic ability of the component to act autonomously in open, distributed and heterogeneous environments, in connection or not with an upper authority. Acknowledging this situation, we are forced to admit that SCADA [4][39] components are no longer to be considered only as basic isolated solution deployed to support business activities, but that they are part of crisis reaction strategy [29]. Since then, acquiring an innovative enterprise architecture framework to represent the behaviors of such components appears fully justified in view of the arising cyber protection principles and required by the practitioners, especially the ones engaged in the management of those critical infrastructures security protection.

In this paper, we propose to explore ArchiMate® and to redesign its structure in order to fit with component software actors' specificities and domain constraints. The main focus concerns the design and the consideration of the policies that are centric concepts related to the activation of component's behaviors. All along the modeling of the SCADA system and the definition of the policies according to these models, we are going to illustrate the theory with a case study related to the petroleum supply chain, and more specially the specific functions of Crude Oil Supply and Crude Oil Storage and Distribution. This extended case is introduced in Section 2. In Section 3, we will review the SCADA components metamodel and the SCADA layers for crisis management and we will model the concept of policy [22][24] that represents the engine of the component modeling framework in Section 4. Section 5 provides related works and Section 6 concludes the paper.

## II. RELATED WORKS

Literatures explain methodologies to model Multi-Agent System (MAS) and their environments as a one layer model and give complete solutions or frameworks. Gaia [1] is a framework for the development of agent architectures based on a lifecycle approach (requirements, analysis, conceptualization and implementation). AUML [6] and MAS-ML [2] are extensions of the UML language for the modelling of MAS but do no longer exist following the release by the OMG of UML 2.0 [11][12] supporting MAS. Prometheus [7] defines a metamodel of the application layer and allows generating organizational diagrams, roles diagrams, classes' diagrams, sequences diagrams and so forth. It permits to generate codes but does not provide links between diagrams and therefore makes it difficult to use for alignment purposes or with other languages (e.g., MOF [3], DSML4MAS [5]). CARBA [15] provides a dynamic architecture for MAS similar to the middleware CORBA based on the role played by the agent. Globally, we observe that these solutions aim at modelling the application layer of MAS. CARBA goes one step further introduces the concept of Interface and Service. This approach is closed to the solution based on ArchiMate® that we design in our proposal but offers less modelling features. As we have noticed that agent systems are organized in a way close to the enterprises system, our proposal analyses how an enterprise architecture model may be slightly reworked and adapted for MAS. Therefore, we exploit ArchiMate®, which has the following advantages to be supported by The Open Group. It has a large community and proposes a uniform structure to model enterprise architecture. Another advantage of ArchiMate® is that it uses referenced existing modelling languages like UML.

## III. COMPLEX OIL DISTRIBUTION TEST MODELING

To represent the modelling of SCADA components metamodel and policy generation, we are going to illustrate this paper with the reference case study presented in [24]. The *crude oil distribution* [22] presented at the Section 2 includes both the oil supply and product distribution SCADA systems. Interconnection amongst Remote Terminal Unit (RTU) [31] of those SCADA is achieved using MTU [30]. The acronym MTU stands for Master Terminal Unit and its main purpose is to accept the different inputs from the remotely connected devices and to transmit these inputs over the rest of the network. Using the *ArchiMate*® for SCADA system theory introduced in previous Sections, the SCADA RTU of the crude oil distribution SCADA from the distributed plants may be modeled as illustrated on Figure 1.

As illustrated in this figure, both layers of the RTU are represented, the COS *SCADA RTU Networks Organization* (RTU-COSNO) and the *COSSCADA RTU Networks Application* (RTU-COSNA). At the COSNO layer, the crude oil network SCADA is composed of Crude oil portfolio that is assigned to *Call for IN* (aka *Organizational alert IN*), of application RTU monitoring services (e.g., *Moni SEGUA* and *Moni SEBAT* (Figure 1)), of RT information that impacts the generation of RTU behavioral policies. On the other side, the SEGUA and the SEBAT RTU (for instance) are represented as *actor* of the RTU organization and are composed of RTU *network console* dedicated to the SCADA management [38]. Both later are associated to the artifact modelled by the orange box that correspond to a collaboration between both SCADA functions, the crude oil supply and the product storage and product distribution. At the COSNA layer, four RTU/technical layer are modelled, respectively the SEGUA, SEBAT, RPBS and REVAP. The structure of this RTU/Technical layer is naturally always quite the same and is composed of the *technical monitoring service* (corresponding to the core of the RTU such as commonly addressed by the literature [21]), and of the interface named "*in-[RTU/network location]*", which aims at connection the monitoring service with the RTU application itself. As illustrated at the level of the SEBAT model, the RTU application is potentially connected with the others RTU's applications artefacts (cf. *two-ring* symbol). As summary, in this Section we have presented a metamodel for SCADA systems. This metamodel allows representing all the components of the SCADA following three layers: the organization, the application and the technical layers. Those models offer the advantages to easily figure out the structure of the concepts and their interconnections and thereby, to easily capture the interconnection between the components within a SCADA and among two or many SCADA's or SCADA functions. Given those advantages, the next Section explains how management policies may be designed and defined according to instance of this SCADA metamodel. Concretely, the usage of the metamodel has been illustrated trough a crude oil supply and distribution plan SCADA and connections have been depicted among the crude oil supply and the crude oil storage and distribution function of the SCADA system.

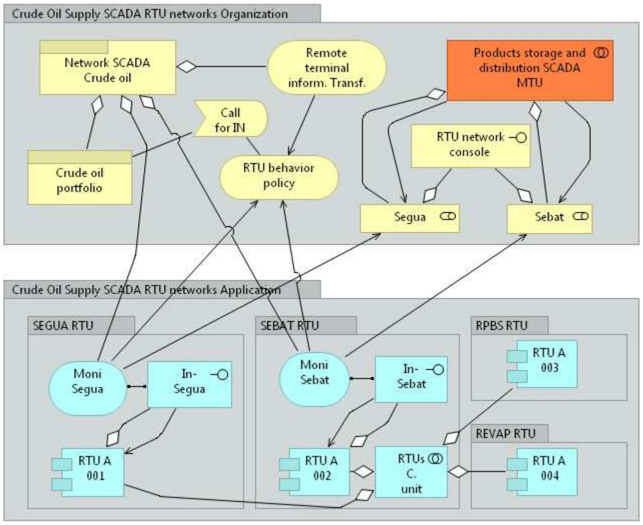

Figure 1. Crude oil SCADA MTU-RTU instance.

## IV. SCADA POLICY MANAGEMENT

Based on *ArchiMate*® SCADA metamodel presented in Section 3 and illustrated by the *Crude Oil Supply* SCADA, this Section introduces the artefact of policy model by ArchiMate. Two types of Policy are depicted: the *Cognitive* and the *Permissive* [12].

### A. *Policy family*

At the *Organizational Layer*, *Policy* can be represented as an *UML Use Case* [11] where concepts of *Roles* represent the *Actors* which have *Responsibilities* in the *Use Case,* and the *Collaboration* concepts show the connections between them. Concepts of *Products*, *Value* and *Organizational Service* provide the *Goal* of the *Use Case*. *Pre-* and *Post-conditions* model the context of the *Use Case* and are symbolized in the metamodel by the *Event* concept (pre-condition) and the *Organizational Object* (pre-/post-condition). In the *Application Layer, Application Policy* is defined as the realization of *Responsibilities* by the *Application Domain* in a configuration of the *Data Domain*. UML provides support for modelling the behavior performed by the *Application Domain* as *Sequence Diagram*. Configuration of the *Data Domain* can be expressed as *Pre-conditions* of the *Sequence Diagram* and symbolized by the execution of a test-method on the lifeline of the diagram. The metamodel designed in Section 3 has allowed providing the SCADA operators and managers with a holistic and integrated view of the SCADA architecture building blocks. In practice and to have policy extracted according to the metamodel concepts interconnections, this SCADA metamodel firstly needs to be instantiated for each architecture components. This step is achieved by shaping the component according to the three abstractions typically advocated by the enterprise architecture paradigm [11][12] and [13]. This allows discovering the building artefacts of the components as well as the connections amongst the components artefacts. An example of this instantiation is represented in Figure 1. The representation of each component implies paramount outcomes for the SCADA [31] operator since it confers to the latter a global functional insight of each component irrespective of any implementation or vendors' influence. The unitary SCADA [28] component models are than used in the second step to picture out the global structure of the SCADA architecture and of the connections, in terms of *policies*, amongst the components of the architecture. Previous works [40] highlights the two families of policies recovered in SCADA [29]: *Permissive policies* and *Cognitive policies. Cognitive Policies (CP)* [12] represent policies which govern the behavior of one artefact of the component architecture. This policy specifies the rule that the *Responsible* artefact needs to follow to execute a defined activity in a specific context. This rule is dictated by the artefact which exists in the same component or in another one. The artefact which generates the policy is the *Master* and the one, which execute it is the *Slave*. The *Cognitive Policy* morphology is articulated on the following set of attributes (perceived by [13]): Master artefact, Slave artefact, Master component, Slave component, Behaving rule, Trigger item, Usage context, Priority extension.

The application schema of a CP, as presented in Figure 1, obeys the two following controls: (1) the communication path is from a *Master* structural concept to a *Slave* behavioral concept or (2) the communication path is from a *Master* behavioral artefact to another *Slave* behavioral artefacts. Figure 2. They represent policies which govern the knowledge acquisition rules from the *Master* to the *Slave* artefact [14]. This knowledge acquisition traditionally takes the form of SCADA states data accessed or provided in order to provide the *Responsible* with the access (of `in`, `out`, `in_out` types [16]) to successive *Cognitive Policies* in case of occurring *events.* The *Permissive Policies* morphology is articulated on the following set of attributes [(perceived by [15]): Master artefact, Slave artefact, Master component, Slave component, Permission rules, Pre-permission conditions, Master permission cardinality, Slave permission cardinality, and Cognitive constraints - sustained by *Cognitive Policy*. The application schema of a CP, as highlighted in Figure 1, obeys the two following controls: (1) the communication path is from a *Master* structural artefact to a *Slave* informational artefact or (2) the communication path is from a *Master* behavioral artefact to a *Slave* informational artefact.

### B. *Policy identification method*

Designing automatic management strategy requires a rigorous two phase's policy elaboration mechanism, respectively the policy scheme identification and the policy scheme formalization.

#### 1) *Policy scheme identification step*

The first step is itself structured in three phases. The first one aims at identifying the structure of the CI architecture in terms of unitary modules (components), including their three layers of abstraction build upon the SCADA [27] metamodel (i.e.,, organization, application, and technical). The second phase aims at identifying the external parameters of the CI (Critical Infrastructure) such as potential threat probes and indicators that may impact the CI normal functioning (flood, hijacking, etc.), the physical environment, and/or the contractual SLA (service level agreement). The third phase aims at identifying the reaction policies which may be of two types: *Cognitive* (artefact of a CI component which needs information from succeeding artefacts –Figure 2) or *Permissive* (artefact of a CI component which needs permission upon the succeeding lower layer artefacts – Red connections on Figure 1). Both types of policies are explained in [35][36][37][20].

#### 2) *Policy scheme formalization step*

After policies being identified, the second step of the method aims at formalizing policy scheme using a three phases approach. The first one aims at depicting the *Master-Slave communication* artefacts (organization-organization, organization-technical, and technical-technical), the second aims at identifying the *cognitive* and *permissive behaviour* based on the automatic reaction strategy, and the last one aims at formalizing the policies accordingly. This latter is function of the policy type and is achieved, on one hand, with the inter-artefacts knowledge requirement, external probes and monitoring tools in case of *cognitive* policy and with the reaction strategy with the requirement of access to artefacts in case of *permissive policy*.

#### 3) *Inter Critical Infrastructures Study Case*

This second part of the case study aims at defining *cognitive* and *permissive* security policies supported by the MTU-RTU model from Figure 1. In [32], authors argue that SCADA system network is *different from general network environment due to its operational environment in national infrastructure*. Therefore, in such a context, the SCADA system needs important *broadcast capability,* which must be highly protected. Among these protection mechanisms are the *key management schemes* [32][33][34] that also have to support the multicasting messages protection. Figure 1 illustrates the modelling of *permissive* and *cognitive* policies related to the Key Management Exchange, such as expressed by [32] among the MTU dedicated to the *crude oil supply* function and the RTU from this function and from the *storage and distribution* function. This field has already been tackled by many researches such as [33] [34]. [32] has been preferred for this case illustration provided that it reduces consistently the number of keys to be stored and provides multicasting and broadcasting communication for efficient and stable operation of SCADA systems. Hence, the policies dedicated to the management of this broadcasting will be defined in the following.

Three constraints related to the key management broadcasting mechanism related to the SCADA architecture have been defined by [18][19][31] and need to be considered along the modelling of the policies: (1) the computational capacity limit which may be represented as an artefact of a type data object at the application layer of the MTU, (2) the low data transmission rate which is also a concept related to the MTU by means of a data object, and (3) the real-time processing that needs to be consider to prevent data processing delay and which may be represented as a data object from the RTUs structures. From Figure 2, we observe the following list of policies: Firstly at the organization layer: the MTU Management policy (1), and secondly at the application layer: the crude oil supply policy /MTU S1 (2) and /RTU S1 (3) and the crude oil storage /RTU Sto1 (4). (1) is existing at the organizational layer and is realized by (2) at the technical layer [17]. This first family of policies (1) accesses the key exchange value that represents the real encryption parameter introduced by the SCADA operator through the dedicated interface (aka MTU screen). The later aims at supporting the key management service which is represented by the key management unit artefact. It has the right of a type in, out, in/out on the key set MTU, key set S1 and key set sto1 data objects (Table I).

TABLE I. PERMISSIVE POLICY FOR ATTRIBUTES' NAME AND ATTRIBUTES' ID

| Attribute Name | Attribute's ID |
|---|---|
| Master artefact | Organizational service |
| Salve artefact | Data objects |
| Master component | Key management unit |
| Slave component | key set MTU, key set S1 and key set sto1 |
| Permission rules | In/Out/In-Out |
| Permission conditions | ∃ of set of Master-Slave Associations |
| Master permission cardinality | 1 |
| Slave permission cardinality | 1..n |
| Cognitive constraints | Key exchange values |

This policy is a *permissive* policy provided that it gives an authorization. The second family of policies depicted through the RTU-MTU model concerns the application layer policies named MTU S1, RTU S1 and RTU Sto1. These policies are directly assigned and dictate the expected behavior of the application function (in this case, the selection of the encryption ID and system). These policies correspond to *Cognitive*. They express that 1 of the MTU S1, RTU S1 or RTU sto1 policy (master artefact) may Select key Encryption ID, May enforce Key Encryption ID and Algorithm [32] related to the application MTU S1, RTU S1 and RTU sto1 (slave artefact) if there exist at least one permission of a type Comp.-capa.-Limit, trans.-rate, real-ti.-proc. To process the above *Cognitive* policies, the MTU S1, RTU S1 and RTU Sto1 policies required to collect information related to the key by directly accessing the respective key set data object artefact, to know: Key Set MTU, Key Set S1 and Key Set Sto1. This collection of information is possible if the appropriate *permissive* policies are defined and deployed in the SCADA. For the sake of clarity, the later have not been represented in the MTU-RTU model (Table II).

TABLE II. COGNITIVE POLICY ATTRIBUTES' NAME AND ATTRIBUTES' ID

| Attribute Name | Attribute ID |
|---|---|
| Master artefact | Application service |
| Slave artefact | Application |
| Master component | Policy MTU S1, Policy RTU S1, Policy RTU Sto1 |
| Slave component | MTU S1, RTU S1, RTU Sto1 |
| Permission rules | Select key Encryption ID – Enforce Key Encryption ID and Algorithm |
| Permission conditions | Comp.-capa.-Limit, trans.-rate, real-ti.-proc |
| Master permission cardinality | 1 |
| Slave permission cardinality | 1 |
| Cognitive constraints | ∃ of Technical MTU S1, RTU S1, STU Sto1. |

## V. EVALUATION

Although the MTU S1 and RTU S1 are SCADA artefacts from the same SCADA (*crude oil supply* SCADA), RTU Sto1 is an artefact from another function, i.e.,: *crude oil storage and distribution*. The later consists in an alternative SCADA system. Using the *ArchiMate*® metamodel for modelling SCADA policies of a type *cognitive* or *permissive* at both the organizational and the technical layers has allowed representing heterogeneous SCADA policies from two different SCADA using the same language (i.e.,: *ArchiMate*® for SCADA systems).

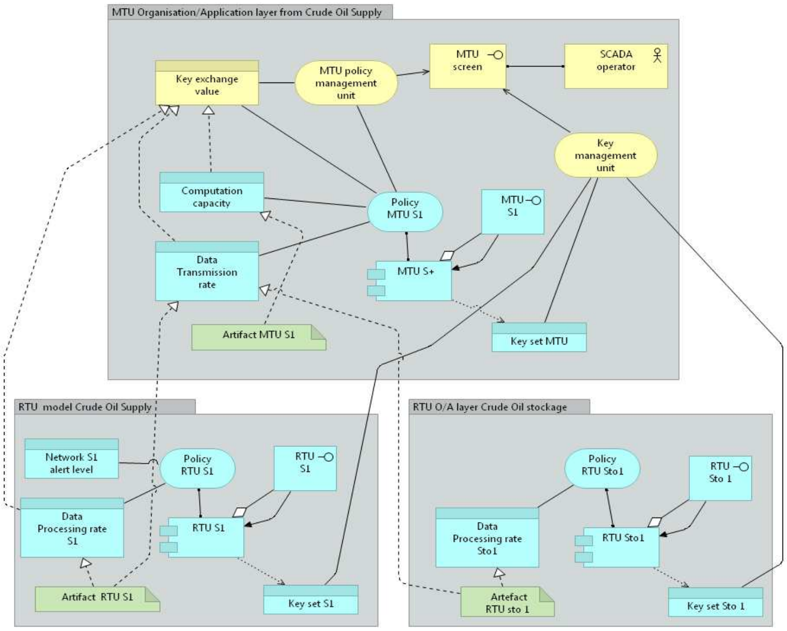

Figure 2. MTU-RTU Key distribution case.

## VI. CONCLUSIONS AND FUTURES WORKS

The paper proposes an integrated approach for modelling the SCADA based on the enterprise architecture modelling language and more specially *ArchiMate®* which has been particularly tailored for SCADA systems [23][24][25][26]. Based on a dedicated metamodel, the paper has demonstrated how technical, application and organization policies could be designed and metamodeled, especially regarding the policy management for interconnected SCADA systems for two of its functions. All along the modelling of the SCADA model and the definition of the policies according to these models, we have illustrated the theory with a business case study related to the petroleum supply chain, and more specially the specific functions of *crude oil supply* and *crude oil storage and distribution*. The main future works consists in elaborating a concrete prototype to sustain the metamodel usage and the deployment in real usage settings. The metamodel drawback concerns the lack of a dedicated specialization of the ArchiMate language. This extension of the framework would highly enrich the decision making mechanism in CI.

## REFERENCES

[1] F. Zambonelli, N. R. Jennings, and M. Wooldridge, 2003, "Developing multicomponent systems: The Gaia methodology". ACM Trans. Softw. Eng. Methodol. 12, 3 (July 2003), 317-370.

[2] V. Torres da Silva, R. Choren, and C. J. P. de Lucena, 2004, "A UML Based Approach for Modeling and Implementing Multi-Component Systems". In Proceedings of the Third AAMAS, Vol. 2. IEEE Computer Society, Washington, DC, USA, 914-921.

[3] J. J. Gomez-Sanz, J. Pavon, and F. Garijo, 2002, "Metamodels for building multi-component systems". Proceedings of ACM symposium on Applied computing (SAC '02). ACM, New York, , USA, 37-41.

[4] G. Beydoun, C. Gonzalez-Perez, G. and Low, B. Henderson-Sellers, 2005, " Synthesis of a generic MAS metamodel". SIGSOFT Softw. Eng. Notes 30, 4, 1-5.

[5] AUML (Component UML), http://www.auml.org/ [accessed: 2015-03-12]

[6] G. Guemkam, C. Feltus, P. Schmitt, C. Bonhomme, D. Khadraoui, and Z. Guessoum, 2011, "Reputation Based Dynamic Responsibility to Agent Assignement for Critical Infrastructure". In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02 (WI-IAT '11), Vol. 2.

[7] C. Feltus, E. Dubois, E. Proper, I. Band, and M. Petit, 2012, "Enhancing the ArchiMate® standard with a responsibility modeling language for access rights management". In Proceedings of the Fifth International Conference on Security of Information and Networks (SIN '12). ACM, New York, NY, USA, 12-19.

[8] Daneels, Axel, and Wayne Salter, "What is SCADA." International Conference on Accelerator and Large Experimental Physics Control Systems. 1999.

[9] C. Feltus, M. Ouedraogo, and D. Khadraoui, "Towards Cyber-Security Protection of Critical Infrastructures by Generating Security Policy for SCADA Systems", The 1st International Conference on Information and Communication Technologies for Disaster Management (ICT-DM'2014), 24-25/3/2014, Algiers, Algeria.

[10] Khadraoui, D., and Feltus, C., "Critical Infrastructures Governance, Exploring SCADA Cybernetics through Architectured Policy Semantic," Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on , pp.4766-4771.

[11] Blangenois, J., Guemkam, G., Feltus, C., and Khadraoui, D., "Organizational Security Architecture for Critical Infrastructure," Availability, Reliability and Security (ARES), 2013 Eighth International Conference on , vol., no., pp.316,323, 2-6 Sept. 2013

[12] C. Feltus, D. Khadraoui, and J. Aubert, "A Security Decision-Reaction Architecture for Heterogeneous Distributed Network". 2012 Seventh Int. Conference on Availability, Reliability and Security. IEEE.

[13] J. Sabater, and C. Sierra, "Review on computational trust and reputation models", Artificial Intelligence Review, vol. 24, no. 1, pp. 33–60

[14] W. Jiao, and Z. Shi; "A dynamic architecture for multi-agent systems", Technology of Object-Oriented Languages and Systems, 1999. TOOLS 31. pp.253-260.

[15] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

[16] J. C. Maxwell, "A Treatise on Electricity and Magnetism", vol. 2. Oxford: Clarendon, 1892.

[17] Davidson, Euan M., et al. "Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data." Power Systems, IEEE Transactions on 21.2 (2006).

[18] C. Feltus, D. Khadraoui, B. de Rémont, and A. Rifaut, "Business Governance based Policy Regulation for Security Incident Response", International Conference on Risks and Security of Internet and Systems 2-5/7/2007, Marrakech, Morocco.

[19] C. Feltus, "Conceptual Trusted Incident-Reaction Architecture", The 6th International Network Conference 2010 (INC2010), June 2010, Heidelberg, Germany

[20] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August, p. 301, 1982.

[21] C. Feltus, "Preliminary Literature Review of Policy Engineering Methods - Toward Responsibility Concept", International Conference on Information & Communication Technologies: from Theory to Applications (IEEE ICTTA2008), Damascus, Syria.

[22] M.S. Neiro, and J. M. Pinto, "A general modeling framework for the operational planning of petroleum supply chains", Computers & Chemical Engineering, Volume 28, Issues 6–7, 2004, Pages 871–896.

[23] C. Feltus, M. Petit, and M. Sloman, "Enhancement of Business IT Alignment by Including Responsibility Components in RBAC", 5th International Workshop on Business/IT Alignment and Interoperability (BUSITAL 2010), 2010, Hammamet, Tunisia.

[24] G. Guemkam, J. Blangenois, C. Feltus, and D. Khadraoui, "Metamodel for Reputation based Agents System - Case Study for Electrical Distribution SCADA Design", 6th ACM International Conference on Security of Information and Networks (ACM SIN 2013), November 2013, Aksaray, Turkey.

[25] Feltus, C. and Petit, M., "Building a Responsibility Model Including Accountability, Capability and Commitment", Availability, Reliability and Security, 2009. ARES '09. International Conference on , vol., no., pp.412,419, 16-19 March 2009.5

[26] Gateau, B., Khadraoui, D., and Feltus, C., "Multi-agents system service based platform in telecommunication security incident reaction," Information Infrastructure Symposium, 2009. GIIS '09. Global , vol., no., pp.1,6, 23-26 June 2009

[27] Patel, S. C., Bhatt, G. D., and Graham, J. H. (2009), "Improving the cyber security of SCADA communication networks". Communications of the ACM, 52(7), 139-142.

[28] Bailey, D., and Wright, E, (2003), "Practical SCADA for industry". Newnes.

[29] Donghyun C.I, Hakman K., Dongho W., and Seungjoo K., "Advanced Key-Management Architecture for Secure SCADA Communications," Power Delivery, IEEE Transactions, vol.24, no.3, pp.1154,1163, 2009

[30] Beaver, C., Gallup, D., Neumann, W. and Torgerson, M. (2002), "Key management for SCADA," Technical report, Sandia.

[31] R. Dawson, C. Boyd, E. Dawson, J. Manuel, and G. Nieto, "SKMA A Key Management Architecture for SCADA Systems," In Proc. Fourth Australasian Information Security Workshop, Vol. 54, pp. 138-192, 2006.

[32] C. Feltus, and D. Khadraoui, "On Designing Automatic Reaction Strategy for Critical Infrastructure SCADA System", 6th ACM International Conference on Security of Information and Networks (ACM SIN 2013), 26-28/11/2013, Aksaray, Turkey.

[33] http://pubs.opengroup.org/architecture/archimate2-doc/

[34] Prometheus Methodology. http://www.cs.rmit.edu.au/agents/SAC2/methodology.html

[35] Chan, M. L. (1991, April), "Interrelation of distribution automation and demand-side management". In Rural Electric Power Conference, 1991. Papers Presented at the 35th Annual Conference (pp. B1-1). IEEE.

[36] Kato, K., and Fudeh, H. R. (1992), "Performance simulation of distributed energy management systems. Power Systems", IEEE Transactions on, 7(2), 820-827.

[37] Choi, D., Kim, H., Won, D., and Kim, S, 2009, Advanced key-management architecture for secure SCADA communications. Power Delivery, IEEE Transactions on, 24(3), 1154-1163.