2015

# Resilient and Trustworthy Dynamic Data-driven Application Systems (DDDAS) Services for Crisis Management Environments

Youakim Badr

Salim Hariti

Youssif AL-Nashif
*Old Dominion University*

Erik Blasch

## Repository Citation

Badr, Youakim; Hariti, Salim; AL-Nashif, Youssif; and Blasch, Erik, "Resilient and Trustworthy Dynamic Data-driven Application Systems (DDDAS) Services for Crisis Management Environments" (2015). *Electrical & Computer Engineering Faculty Publications*. 128.
https://digitalcommons.odu.edu/ece_fac_pubs/128

## Original Publication Citation

Badr, Y., Hariri, S., Al-Nashif, Y., & Blasch, E. (2015). Resilient and trustworthy dynamic data-driven application systems (DDDSS) services for crisis management environments. *Procedia Computer Science, 51*, 2623-2637. doi:https://doi.org/10.1016/j.procs.2015.05.370

Procedia Computer Science

Volume 51, 2015, Pages 2623–2637

ICCS 2015 International Conference On Computational Science

# Resilient and Trustworthy Dynamic Data-Driven Application Systems (DDDAS) Services for Crisis Management Environments

Youakim Badr
Université de Lyon, CNRS
LIRIS, UMR5205, INSA-Lyon, France

Salim Hariri
*Center for Cloud and Autonomic Computing*
*The University of Arizona, Tucson, Arizona*

Youssif AL-Nashif
Old Dominion University, Norfolk, Virginia

Erik Blasch
*Air Force Research Laboratory, Information Directorate, Rome, NY*

**Abstract**
Future crisis management systems need resilient and trustworthy infrastructures to quickly develop reliable applications and processes, and ensure end-to-end security, trust, and privacy. Due to the multiplicity and diversity of involved actors, volumes of data, and heterogeneity of shared information; crisis management systems tend to be highly vulnerable and subject to unforeseen incidents. As a result, the dependability of crisis management systems can be at risk. This paper presents a cloud-based resilient and trustworthy infrastructure (known as rDaaS) to quickly develop secure crisis management systems. The rDaaS integrates the Dynamic Data-Driven Application Systems (DDDAS) paradigm into a service-oriented architecture over cloud technology and provides a set of resilient DDDAS-As-A Service (rDaaS) components to build secure and trusted adaptable crisis processes. The rDaaS also ensures resilience and security by obfuscating the execution environment and applying Behavior Software Encryption and Moving Technique Defense. A simulation environment for a nuclear plant crisis management case study is illustrated to build resilient and trusted crisis response processes.

*Keywords:* rDaaS, resilience, cloud computing, service-oriented computing, trust and security

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.370

# 1 Introduction

Crisis management (CM) for critical infrastructures, natural disasters, or civil unrest caused by natural or malicious forces requires complex, dynamic and heterogeneous responses. Distributed CM systems should detect, recognize, and disseminate huge amounts of heterogeneous dynamic events that operate at different speeds and formats. Furthermore, the processing of crisis events and the development of real-time responses are challenges when the security and trust of every human and resource must be evaluated to maintain security, privacy and trust among different stakeholders collaborating in the crisis management environment. For example, managing a nuclear crisis requires several actors to establish *response crisis processes* to coordinate their missions and securely exchange information based on their trusted levels. Multiplicity and diversity of involved actors, volume and heterogeneity of shared information, critical dependencies between actions, as well as the unpredictability of the situation evolution; make the crisis environment complex and dynamic. In this paper, we present our approach to leverage Dynamic Data-Driven Application Systems (DDDAS) techniques, Service-Oriented Architecture (SOA) paradigm, and cloud services to develop a trusted crisis management environment that we refer to as resilient DDDAS-As-A-Service (rDaaS)..

Service-oriented computing aims at assembling software components to build distributed, agile applications and business processes that span organizational, communications and computing platforms. Despite the agility that service-oriented architectures promise, many SOA-based processes are increasingly encountering difficult self-adaptation issues and fail to deal with continuous and unpredictable changes in dynamic environments. Changes may occur during runtime due to unavailable services or unreachable actors, resources degradation and security threats just to mention a few. Changes may also evolve from unforeseen incidents and new decisions that, consequently, require rebuilding responses processes partially or fully. On the other hand, the DDDAS paradigm allows continuous monitoring, analysis and adaptation of all software components involved in networked systems, including sensors, computational resources and applications. In addition, cloud computing is particularly a promising technology that may support crisis management with computational and storage capabilities and quickly deploy crisis processes. Virtualization and elasticity of resources makes possible to deploy application servers and devices, and scale-up with large amount of data collected from various sources. Consequently, the integration of DDDAS and SOA into one cloud environment, called rDaaS, leads to a novel architecture enabling the construction of adaptable SOA-based processes to deal with unpredictable changes in dynamic environments.

In this paper, we present a DDDAS cloud-based resilient and trustworthy infrastructure to quickly develop crisis management applications and processes, and robustly tolerate attacks against cloud systems and services. Processes are built based on resilient SOAs driven by requirements and adaptable to internal and external changes. In rDaaS, the cloud environment tends to continuously design, develop and execute crisis management response processes and services. In order to ensure resilience and security against intrusions, the rDaaS continuously obfuscates the execution environment using Behavior Software Encryption, Moving Technique Defense (MDT) and autonomic management. As a result, rDaaS applications and services run diversified versions on replicated hardware, change randomly their implementation and resources, and consequently evade attackers. To enable trusted collaboration among different actors involved in the crisis management environment, the rDaaS evaluates trust values of all actors and resources at runtime before enabling trustworthy collaborations based on different trust levels.

The remaining sections of the paper are organized as follows. Section 2 describes related work and background research. Section 3 describes the Cloud-based resilient and trustworthy Dynamic Data-Driven Application Systems for Crisis Environments (called r-DDDAS-as-a-Cloud Service (rDaaS)) to develop resilient crisis management applications. Section 4 presents our experimental design and evaluation of the rDaaS approach. Finally the paper concludes in Section 5.

# 2 Related Work

Crisis management has received an abundant attention from a managerial perspective as they are often reduced to suitable strategies to prepare for handling the crisis, ensure a rapid and adequate response, and define clear roles and responsibilities. Critical infrastructure designs require cybersecurity (trust) which includes adaptability by continuously integrating design and runtime (design once – run forever) support. In this section, we will review the cybersecurity detection techniques in general and then give a brief overview of resilient techniques.

## 2.1 Security Related Work

Intrusion detection can be broadly classified as signature-based and anomaly-based systems.

A *signature-based Intrusion Detection System* (IDS) uses pattern-matching algorithms to compare network traffic with an extensive library of attack signatures created by human experts [1]. A match indicates a probable attack. These techniques are extremely proficient in detecting known attacks because they can identify an attack as soon as it occurs. However, their foremost limitation is that it cannot detect new attacks. When a new attack is discovered, it takes time to develop signatures for the attack and deploy it into the existing IDSs. Some of the most commonly used signature-based intrusion detection techniques are introduced by SNORT [2], BRO [3], and others [4]. Anomaly-based detection techniques build a model of normal behavior and automatically classify statistically significant deviations from the normal profile as being abnormal [5], [6], [7], [8]. The advantage of this approach is that it is possible to detect unknown attacks. However, there is a potential for having a high rate of false positive alarms generated when the knowledge collected about normal behaviors is inaccurate. Examples of such techniques include Anomaly Behavior Analysis (ABA) methodology that has been successfully applied to a wider range of protocols (TCP/IP, DNS, WiFi, HTTP, etc.) [5] [9] [10] [11], IDES [12], NIDES [13], EMERALD [14], and SPADE [15].

An *anomaly-based IDS* builds a pattern based on a normalcy model and the compares incoming data to the trained model [16-25]. The construction of the pattern, model, or database of normalcy can be coordinated with data mining techniques and machine learning [26]. For situations in which the actors are dynamic, game-theoretic approaches for cyber security over IDS incorporates learned behaviors for the normalcy of the situation [27]. The anomaly can be a threat to standard operations either measured or predicted [28].

Cloud security suffers from a wide range of attacks such as those that target physical machines as well as the cloud virtualized environment [29]. The dependency of cloud computing on the virtualized environment raises more security issues, like hypervisor exploitations [30] [31]. In addition, one of the main security issues in cloud computing is the insider attacks. With exchange of cloud data between different organizations, the risk of insider attacks increases.

Some previous works have presented classifications of Cloud Security [32-34]. Cross-site scripting [35], Access control weaknesses, OS and SQL injection flaws, cross-site request forgery [36], cookie manipulation, hidden field manipulation, insecure storage and insecure configuration are the threats to data stored in a SaaS cloud [37]. Since, in the cloud model, the customers' data reside on the third-parties' data-centers, data security is a major concern for cloud consumers and providers and some researchers have addressed data security in their works [38-40]. Virtualization is one of the fundamental concepts of cloud computing. In a cloud system, multiple virtual guest machines share the same resources of a physical host machine. Recent advances have sought to combine cloud computing elasticity [41], DDDAS [42], and information fusion [43], for secure applications. There are some known security issues with common VMMs (e.g. Xen, Microsoft HyperV) which can be exploited to threaten cloud services [44].

## 2.2 Resilient Related Work

Moving Target Defense has been identified as a game changer approach to build self-defending systems [45]. Some works presented a wide range of Moving target Defense (MTD) techniques to continuously change network configurations or parameters, firewall settings, operating systems, memory addresses, instruction sets, or application execution environments [46,47]. For example, in [46], the IP addresses are dynamically changed while maintaining existing connections. One can also randomize the configuration [48] where the configuration variables of a system are randomized, while ensuring the availability of end to end services. In [49], the authors presented a survey of several software fault tolerance techniques. The fault tolerance techniques that are based on diversity include dual-node redundant operating stations with hardware or software result comparison, Recovery Block Station [50], Distributed Recovery Block with acceptance Test [51], Voting Triple Modular Redundant Computing Stations [52], N version programming [53], and an Integrated Voting Algorithm [54]. Also, in [55], several diversity defense techniques in popular operating systems were discussed. These include Address Space Randomization [56], Instruction Set Randomization [57], and Data Randomization [58]; where a comparison of these techniques is shown in [59].

Some previous works have adopted diversity as a defense technique in a cloud environment. In [60] the authors envision a Cloud of Clouds Architecture, which provides incrementally high levels of security and dependability to cloud infrastructures, in an open, modular and versatile way. Their architecture employs diversity in deployment of cloud alternatives. However, they do not employ shuffling on these alternatives. In [61], a framework for Proactive Fault Tolerance is discussed that predict failures in nodes and migrate their processes away from the nodes that are about to fail. In [62], the authors envision a cloud environment with continuous change in system configuration in order to create an unpredictable target for an adversary. In [63], the authors presented an intrusion tolerant cloud architecture that adopts the method of hybrid fault model, active and passive replicas, state update and transfer, proactive recovery and diversity. Future methods will use the MapReduce for cloud security for various crisis management applications such as object detection [64], movement [65], and cybersecurity [66].

In our approach to implement rDaaS, we adopt diversity technique to the cloud application execution environment, redundancy in the resources is used to run the cloud services and randomly changing the versions, and resources are used to make it prohibitively expensive for attackers to figure out current execution environment, succeed in exploiting vulnerabilities, and launch attacks.

# 3 r-DDDAS-as-a-Cloud Service (rDaaS) architecture

The r-DDDAS-as-a-Cloud Service (rDaaS) architecture aims at aligning cloud technology with the DDDAS paradigm. Figure 1 shows rDaaS main cloud services, which are used to develop resilient and trustworthy cloud-based event-driven crisis management systems. In this architecture, we leverage the DDDAS paradigm to combine the design stage with the runtime stage. The inseparability between design time and runtime makes it possible to deal with unpredictable events and enable prompt responses to confine and manage mitigation activities.

At the *design stage*, the resilient SOA Editor (RSE) helps decision makers to express their response plans in an abstract manner and to develop effective and trusted and resilient responses even when they are experiencing cyber terrorisms or attacks. Likewise, the editor supports the collection and management of information for decision makers [67, 68]. Information fusion techniques can be supported for DDDAS-based systems using cloud services [69] for UAV control [70]. The RSE generates SOA-based executable processes from abstract response processes, taking into account global constraints expressed in contextual models (see Figure 1).

At the *runtime stage*, the Crisis Runtime Manager (CRM) transforms executable response services into equivalent resilient and trustworthy services using the Resilient Cloud Middleware (RCM). In what follows, we discuss in further details the RSE, the CRM and the RCM functionalities.

.

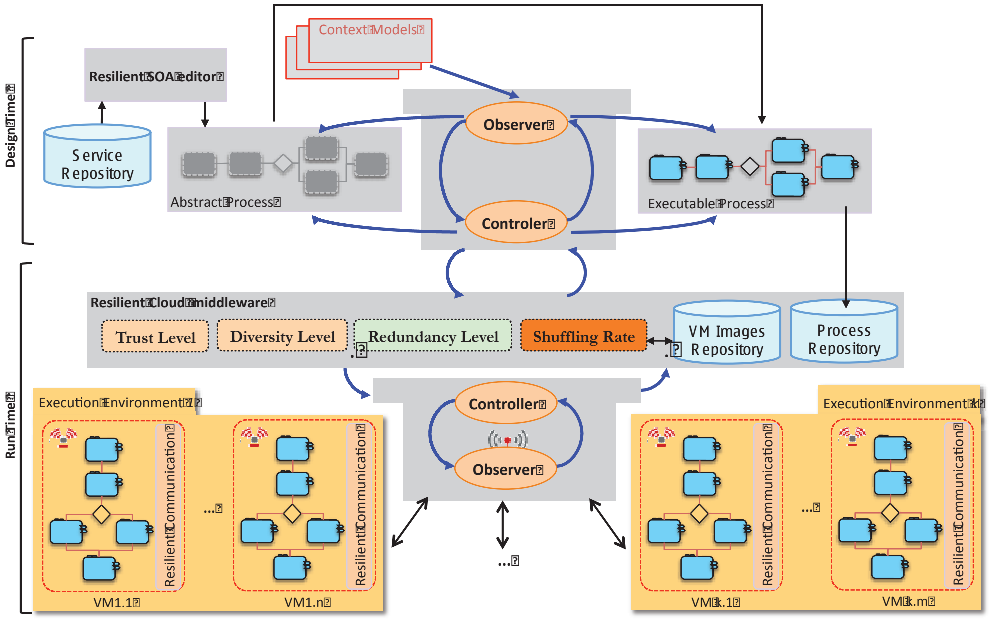


**Fig. 1** rDaaS: Cloud-based resilient and trustworthy Dynamic Data-Driven Application Systems

## 3.1 Resilient SOA Response Service Design

The *resilient SOA Editor* (RSE) is a design assistant editor that assists decision makers to specify at the abstract level how the crisis response will be carried out by using several services provided by the SOA environment as shown in Figure 2. Services, $S_i$ $\{i = 1, \ldots, n\}$ help crisis teams to communicate and interoperate with different types of internal and external resources. Examples include the integration of voice with imagery systems [71, 72].

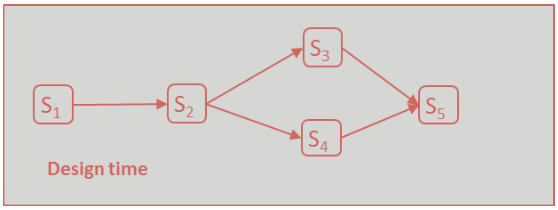


**Fig. 2** Abstract Response Process

In addition, the crisis coordinator can specify resilient requirements of services as well as the required trust level for all the actors involved in the execution of the crisis response services. The resilient requirement can be characterized by: 1) defining the required diversity level (how many different versions of an application and/or how many different platforms (e.g., operating system types) that are required to run the application; 2) defining the redundancy level (e.g., how many redundant physical machines are required); and 3) defining how often the execution environment needs to be changed (e.g., the number of application execution phases).

By using the resilient SOA Editor and the Resilient Cloud Middleware, we can continuously adopt the design of the crisis responses to match the security/trust policy, the context models, and any changes external or internal to the environment as shown in Figure 3.

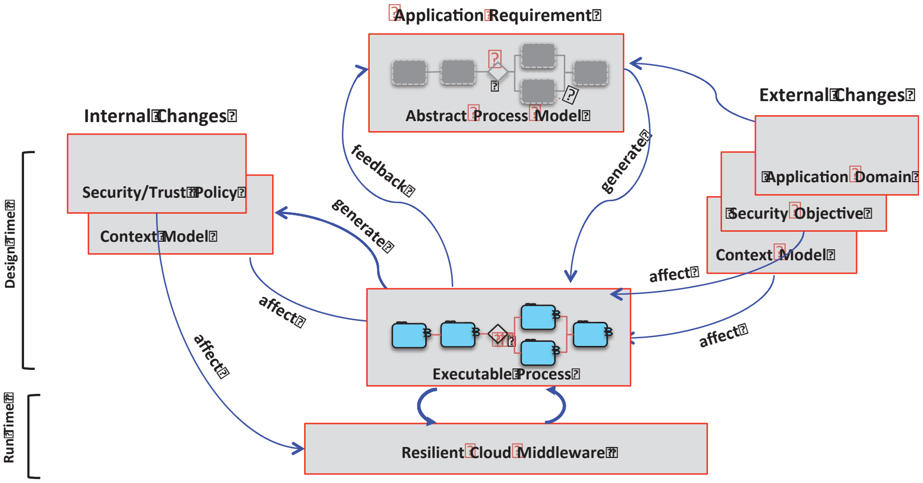


**Fig. 3** Integrating design development stage of crisis response services with their runtime execution environment.

The resilient SOA Editor's implementation is based on the *service farming–based composition* algorithm, which is developed by W. Li [73]. The service-farming algorithm generates optimal executable Web service-based processes from an abstract process by satisfying constraints on Web services (functional and non-functional properties) and models that describe internal and external changes (Figure 3). Abstract processes are written with Business Process Model and Notation (BPMN) [74] to describe global functionalities independent from service implementations whereas models of external and internal changes are expressed with declarative rules that should be satisfy by the service farming algorithm and controlled by the Crisis Runtime Manager.

## 3.2 Crisis Runtime Manager

The Crisis Runtime Manager (CRM) provides control and management services to deploy and configure crisis response services and hardware resources that are required to achieve the response resilient requirements as specified by the Crisis Design Assistant. The CRM is implemented using our autonomic computing architecture shown in Figure 4.

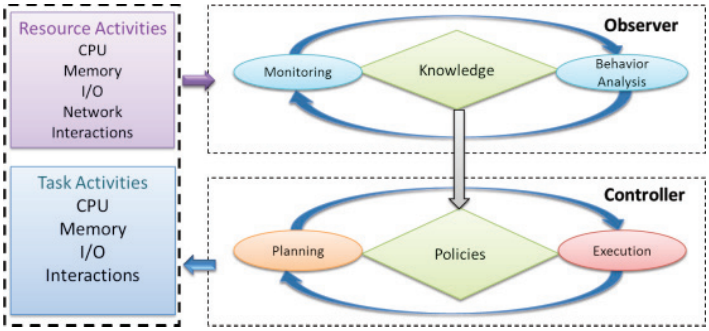


**Fig. 4** CRM Architecture.

The CRM implementation is based on the autonomic computing environment (Autonomia) developed by the UA researchers [75, 76]. The CRM functions will be implemented using two software modules: Observer and Controller modules. The Observer module monitors and analyzes the current state of the managed cloud resources or services. The Controller module is delegated to manage the cloud operations and enforce the resilient operational policies. In fact, the Observer and Controller pair provides a unified management interface to support the CRM's self-management services by continuously monitoring and analyzing current cloud system conditions in order to dynamically select the appropriate plan to correct or remove anomalous conditions once they are detected and/or predicted. The Observer monitors and gathers data about the cloud resources and analyzes them to form the knowledge required by the Controller to enforce the ideal security/resilient management policies.

## 3.3 Resilient Cloud Middleware

The Resilient Cloud Middleware (RCM) provides tools and algorithms to implement desired resilient services as specified by the resilient SOA Editor. These capabilities are briefly described below:

- **Replication/Redundancy:** Redundancy is an approach that applies duplicated versions of the original system to tolerate hardware/software failures in one of the replicated components or systems. Redundancy can be implemented by using hardware, information, time or software redundant techniques.
- **Diversity and Automatic Checkpointing**: This capability enables us to generate multiple functionally-equivalent, behaviorally-different software versions (e.g., each software task can have multiple versions, where each version can be a different algorithm implemented in different programming language (e.g., C, Java, C++, etc.) that can run on different computing systems.
- **Behavior Obfuscation (BO):** The BO method uses spatiotemporal behavior obfuscation to make active software components change their implementation versions and resources continuously and consequently evade attackers. The BO approach will significantly reduce the ability of an attacker to disrupt the normal operations of a cloud application. Also, it allows for adjusting the resilience level by dynamically increasing or decreasing the shuffling rate and scope of executing tasks' versions and their execution environments. A major advantage of this approach is that the dynamic change in the execution environment will hide the software flaws that would otherwise be exploited by a cyberattacker.
- **Trust Level.** The middleware provides the required algorithms to evaluate the trust of all resources and actors involved in the crisis management processes. When an entity communicates with another entity, the CRM obtains the trust level of the entity that needs to interact with from a Trust Authority (TA), see Figure 5. If the trust level of the remote entity is below the minimum required trust level set in the policies, then the communication is dropped. By continuously checking with TA module, any interacting entities will not be able to communicate if they do not meet the end-to-end trust policies. Once the component trust level is verified, they can proceed and interact securely using the secure communications. For further details about our DDDAS approach to evaluate and adopt the trust, please refer to our paper [77].

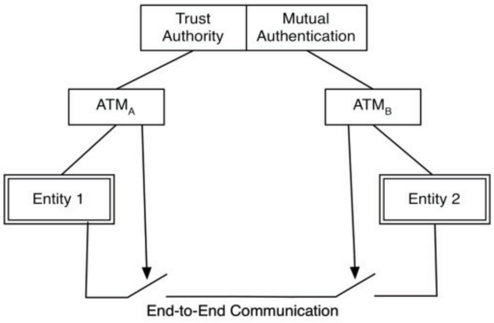


**Fig. 5** Adaptive End-to-End Trust

The resilient operation for any cloud service is achieved using RCM tools will enable us to use the Behavior Obfuscation (BO) algorithm that hides (analogous to data encryption) the execution environment by dynamically changing the number of versions used to run the service at each phase. The dynamic change in the service behavior makes it extremely difficult for an attacker to generate a profile with the possible flaws in the current implementation of the service. The decisions regarding when to shuffle the current variant, the shuffling frequency, and the variant selection for the next shuffle are guided by a continuous monitoring and analysis of current execution state of cloud applications and the desired resilience requirements. In addition to the shuffling of the execution of different application versions, we also apply hardware redundancy in order to tolerate the case when an insider discovers one of the physical machines used to run the application at one phase. The redundancy level determines how many of such scenarios can be tolerated. To speedup the process of selecting the appropriate resilient algorithms and execution environments, the CRM repository contains a set of BO algorithms and images of virtual machines that run in different operating systems (e.g., Windows, Linux, etc.) to implement supported crisis cloud services.

As an illustrative example, let us assume that during the design stage, the decision maker develop a crisis response that consists of five services that were verified by the RSE module to be effective and meets the security and objective requirements. Once that is done, it will be passed to the CRM module that will use the RCM services and tools to build resilient execution of the selected services at runtime as shown in Figure 6 that shows how the versions to be implement at each service at runtime.

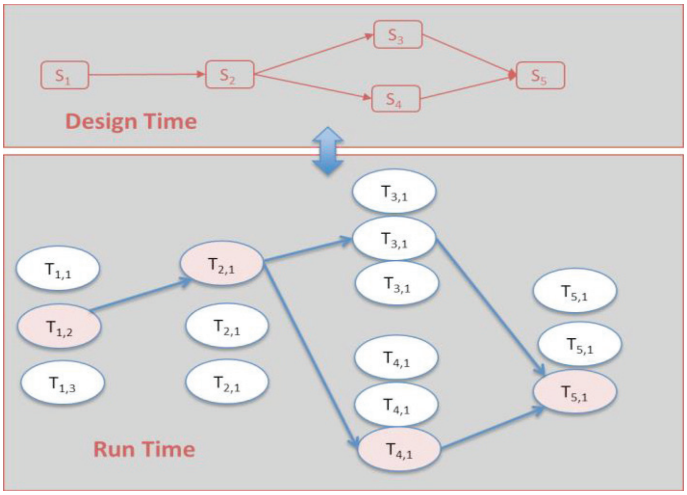


**Fig. 6** Example implementation of five resilient crisis response services

# 4 Case Study: Nuclear Plant Crisis Management

In order to experiment the Dynamic Data-Driven Application Systems (DDDAS) paradigm in managing crisis, we develop a crisis case study in which a large quantity of radioactive substance is accidentally evaded due to a critical incident in a nuclear plant. In response to this crisis, various parties at different levels and responsibilities have to collaborate together to reduce contamination risks. The volume and heterogeneity of exchanged information and critical dependencies between actions make crisis response processes vulnerable and subject to changes.

We build our crisis management system (CMS) based on the nuclear crisis management simulation in the SocEDA project in which eighteen processes have been developed to validate a Cloud-based event-driven architectures [78]. As illustrated in Figure 7 (left), the CMS involves identifying, assessing, and handling the crisis situation by orchestrating the communication between all actors involved in the crisis. It allocates and manages internal and external resources, and interoperates with different external services (e.g., military systems, police systems, government, medical services, etc.). It also provides access to authorized parties to exchange relevant information based on their trust levels. In our scenario, a crisis is usually triggered by report from a witness and/or the surveillance system. A coordinator, who organizes all required resources and tasks, initiates the crisis response process. Observers with expertise in the nuclear field are assigned to the scene to observe the emergency situation. The tasks defined by the observer are crisis missions need to be processed to cope with the situation. Based on observer feedbacks from the scene, the coordinator is required to allocate suitable resources to each task. Human resources, for instance firemen, doctors, policemen, and technicians, acts as first-aid workers, and resilient infrastructure, which may include resilient cloud resources and communication devices (e.g., PDAs or mobile phones). The workers are expected to perform their assigned tasks and report on the success or failure in carrying out the missions that allow the crisis to be concluded. Crisis response processes thus consists of tasks that should be executed in a specific orders to coordinate the communication between involved parties.

All crisis response processes should also keep track of trusted workers and enable rescue resources to provide or access location-sensitive information on the move. Figure 7 (right) shows a three-level mechanism (white, black and grey boxes) to indicate the trust level for accessing information with different trust levels (e.g., detailed maps, terrain data and weather conditions) and routes leading to it.

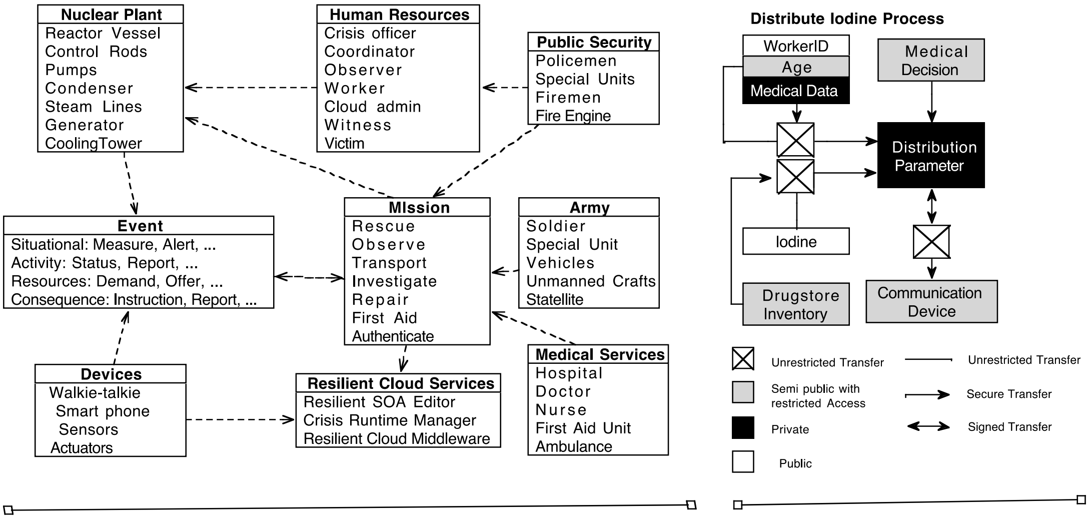


**Fig. 7** Nuclear Crisis Feature Diagrams and Trust Management

As depicted in Fig. 8, the crisis response process involves three levels of tightly integrated functions: 1) *Strategic Level objectives* that will focus on protecting citizens, managing the situation, and recovering using the support of experts and scientific community; 2) *Operational Level tasks* that focuses on distributing, evacuating and confining activities and managing alerts and media; and 3) *Tactical Support Level situational assessments*, for managing resources and data (e.g., information fusion). These objectives, tasks, and assessments need to be performed as resilient and trusted cloud services such as Capture Incident, Assign Internal Resource, Assign External Resource, Execute Missions, Execute Observer Mission, Perform Rescue Missions, and Manage Adaptive end-to-end Trust. Details of tasks related to unforeseen situations (i.e., severe weather, risk of explosion, security attacks, etc.) that may affect the context in which the processes operates, and that require the adaptation to the environment are not included for space reasons. Keeping in mind that the rDaaS architecture deals with the environmental contextual changes and reacts in a certain way to ensure reliability and trustworthy.

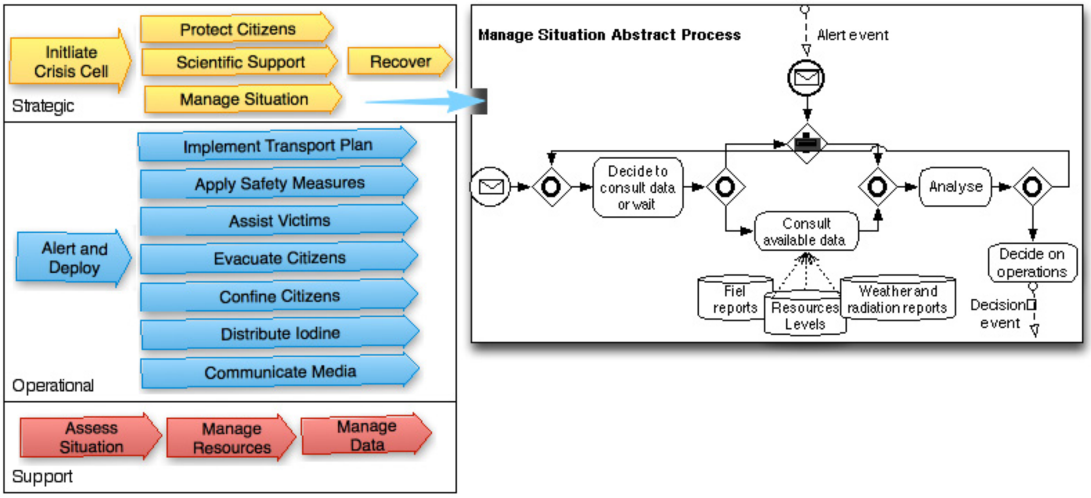


**Fig. 8** Abstract Crisis Response Processes (based on [70])

The crisis management system contains server clusters and middleware that are hosted by the cloud and connected with different critical infrastructures, machines, and crisis team stakeholders. The crisis team members, such as coordinators and cloud administrators, use terminals or desktop machines to access the resilient Cloud services. External services and mobile workers with laptops are also connected to the cloud by means of Web services. The simulation of our CMS consists of:

1- The *simulated environment*, which comprises crisis response processes implemented as Web Services, each of which simulates the evolution of the crisis situation (e.g., the radiation propagation) or to receive feedbacks (events) from the crisis team (e.g., the evacuation status);

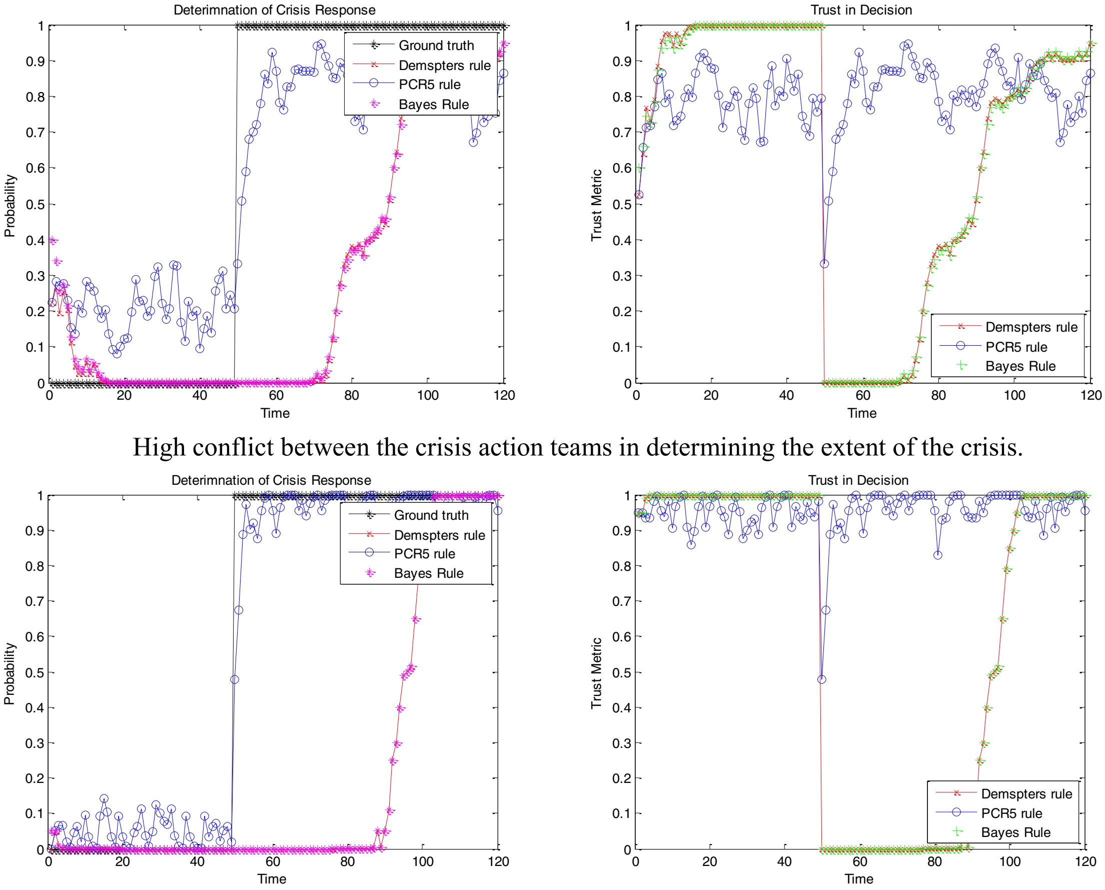
2- The *resilient SOA Response Service Development*, which enables the design of abstract processes and their implementation through distributed ESBs (Enterprise Service Bus). The ESB is used to simulate interactions between crisis actors by invoking Web service–based processes written in BPEL (Business Process Execution Language) and executed through a BPEL engine; and

3- The *Cloud-based Resilient and Trustworthy infrastructure* provides services to access the resilient cloud middleware, ensure security and establish end-to-end trust among various actors.

Since crisis management processes are event-driven, they particularly require an event-driven SOA (a.k.a SOA 2.0) that integrates complex event processing (CPE) and ESB. We use the Synapse/WSO2

ESB as a smart event broker that integrates events from Web services and re-distributes them to end-points (e.g., actuators, terminals). The Sci-Flex plug-in makes possible to connect the Synapse with the Esper, a complex event processing (CEP). Esper rapidly processes large amount and high-frequency time-based event data. The Synapse/WSO2 ESB implements the WS-Eventing to provide simple publish-subscribe model in Web services and event subscription over SOAP. In our case-study, the ESB is considered as event consumer and Web Services are considered as event sources. An event is expressed in the WS-Eventing format and has an event type, which is part of an event topic.

In our case study, we focused on the development of the design environment in order to record and play streams of events and capture them by the event-driven SOA and processes. Interested readers can refer to our previous work on resilient cloud services using DDDAS and moving target defense in [74-77]. The metric-based evaluation regarding the simulation of our event-driven SOA in the resilient cloud supports trust in analysis. For crisis response, situation awareness is needed [79] and more importantly trust in the data being circulated as to an emergency situation. Current challenges in managing the information is the presentation of the data to the user in normal and conflicting scenarios [80, 81]. We chose to use the proportional conflict redistribution (PCR) rule [82] as compared to Bayesian and Dempster-Shafer rules. The uncertainty analysis using PCR has supported trust related metrics (for more detail see [83-86]) which supports timely mission response. Figure 9 shows the case for high conflict (top) and low conflict (below) between crisis action teams.



High conflict between the crisis action teams in determining the extent of the crisis.

Low conflict between the crisis action teams in determining the extent of the crisis.

**Fig. 9** High and Low Data Conflict Between Crisis Action teams for timely response.

The difference between the scenarios which have high and low conflicts between information sharing has a big impact on the system trust. In Figure 9 at the top, there is *high conflict* as different groups are unsure about whether a crisis is happening. The time to make a decision (say above 60%) is not till (90s – 40s) = 50s after the crisis occurred for Bayesian and DS methods. PCR detects the conflict immediately but has a 5-10s delay.

The trust metric is more revealing in the *low conflict* scenario. As with high conflict, the PCR evidential reasoning approach has a quicker reaction, while the Bayesian/DS have slightly longer delay times. The low conflict actually causes the processing steps to become complacent with large biases towards the status quo. Trust metric is higher for all methods in the low-conflict scenario; however, the PCR method stays at a reasonable high trust value throughout the crisis. Thus, variations in the scenario impact the trust metric results, but the higher trust is consistent with lower conflict. The higher trust has demonstrated a measure of resiliency in the data processing for emergency response situations which can be used in context modeling [87].

# 5 Conclusions and Future Work

Future trends for the integration of DDDAS and SOA into one cloud environment include streaming big data, distributed information fusion, and detection of contextual changes in dynamic environments. In this paper, we demonstrated the benefits of DDDAS to enable resilience at the SOA-based process level by combining the design time and runtime process to enable end-to-end trusted and resilient services. Using the simulation framework for nuclear crisis management systems, we are conducting experiments and comparison with existing nuclear crisis management systems. We are also exploring rDAAS with a variety of multi-modal sensing and information fusion services to handle partial or missing contextual information while preserving SOA adaptability to changes and unforeseen situations. We are working on a simulation environment that record and play streams of events with temporal constrains that can extend to other dynamic situations.

## Acknowledgements

This work is partly supported by the Air Force Office of Scientific Research (AFOSR) Dynamic Data-Driven Application Systems (DDDAS), AFOSR **FA95550-12-1-0241**, National Science Foundation NSF IIP-0758579, NCS-0855087 and IIP-1127873, and Thomson Reuters in the framework of the Partner University Fund (PUF) project (PUF is a program of the French Embassy in the United States and the FACE Foundation and is supported by American donors and the French government).

# References

[1] D. L. Pipkin. *Information security: protecting the global enterprise*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2000.

[2] M. Roesch, “Snort - Lightweight Intrusion Detection for Networks,” *13th Systems Administration Conference - LISA* 1999.

[3] V. Paxson, “Bro: a system for detecting network intruders in real-time,” *Computer Networks* (Amsterdam, Netherlands: 1999), 31(23–24):2435–2463, 1999.

[4] K. Scarfone and Peter Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*. Computer Security Resource Center (National Institute of Standards and Technology) (800–94), 2007, Retrieved 1 January 2010.

[5] Y. B. Al-Nashif, A. Kumar, S. Hariri, Y. Luo, F. Szidarovszky, and G. Qu: "Multi-Level Intrusion Detection System (ML-IDS)," *ICAC*: 131-140, 2008

[6] L. Ertöz, E. Eilertson, A. Lazarevic, P. Tan, V. Kumar, J. Srivastava, *et al*., "Minds - Minnesota intrusion detection system," *Data Mining-Next Generation Challenges and Future Directions*. MIT Press, 2004.

[7] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Software Engineering*, 13(2):222–232, 1987.

[8] H. S. Javitz and A. Valdes, *The nides statistical component: Description and justification*, Technical Report, SRI International Menlo Park, California, 1994.

[9] H. Alipour, Y. Al-Nashif, S. Hariri, "DNS Anomaly Behavior Analysis against Cyber Attacks", submitted to *ACM Transactions on Internet Technology*.

[10] H. Alipour, Y. Al-Nashif, S. Hariri, "IEEE 802.11 Anomaly Behavior Analysis", submitted to *IEEE Transactions on Information Forensics and Security*.

[11] R. P. Viswanathan, Y. Al-Nashif, S. Hariri, "Application Attack Detection System (AADS): An Anomaly Based Behavior Analysis Approach," *ACS/IEEE Int'l Conf. On Computer Systems and Applications*, 2011.

[12] T.F. Lunt and R. Jagannathan, "A prototype real-time intrusion-detection expert system", In Proceedings of the *IEEE Symposium on Security and Privacy*, pp. 18–21, 1988.

[13] D. Anderson, T. F. Lunt, H. Javitz, A. Tamaru, and A. Valdes, "Detecting unusual program behavior using the statistical component of the next-generation intrusion detection expert system nides", Technical Report *SRI-CSL-95-06*, Computer Science Laboratory, SRI International, 1995.

[14] P.A. Porras and P.G. Neumann, "Emerald: Event monitoring enabling responses to anomalous live disturbances," In *Proceedings of the National Information Systems Security Conf.*, pp. 353–365, 1997.

[15] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *J. Computer Security*, 10(1-2):105–136, 2002.

[16] K. Sequeira and M. Zaki, "Admit: anomaly-based data mining for intrusions," Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp. 386–395, 2002.

[17] N. Ye, "A markov chain model of temporal behavior for anomaly detection," In *Proceedings of IEEE 5th Annual IEEE Information Assurance Workshop*. 2004.

[18] K. Yamanishi, J. Takeuchi, G. J. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Knowledge Discovery and Data Mining*, pp. 320–324, 2000.

[19] N. Ye and Q. Chen, "An anomaly detection technique based on a chi-square statistic for detecting intrusions into in-formation systems," Quality and Reliability Engineering International, 17, 105-112, 2001.

[20] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," In *Proc. of Applications of Data Mining in Computer Security*. Kluwer Academics, 78-100, 2002.

[21] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *In SIGMOD Conference*, 2001.

[22] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," *Proc. of ACM SIGMOD International Conference on Management of Data*. pp. 93–104, 2000.

[23] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets", In Proc. 24th *Int. Conf. Very Large Data Bases*, VLDB, pages 392–403, 24–27 1998.

[24] S. Ramaswamy, R. Rastogi, and K. Shim. "Efficient algorithms for mining outliers from large datasets," *Proc. ACM SIDMOD Int. Conf. on Management of Data*, pp. 427–438, 2000.

[25] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting network intrusions.", In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 265–274, New York, NY, USA, 2002.

[26] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Science*, 177(18):3799–3821, 2007.

[27] D. Chen, G. Chen, J. Cruz, L Haynes, *et al*., "A Markov Game Theoretic Data Fusion Approach for Cyber Situational Awareness," *Proc. of SPIE*, Vol. 6571, 2007.

[28] G. Chen, D. Shen, C. Kwan, J. Cruz, *et al*., "Game Theoretic Approach to Threat Prediction and Situation Awareness," *Journal of Advances in Information Fusion,* Vol. 2, No. 1, 1-14, June 2007.

[29] "Security as a Service," Cloud Security Alliance, [Online]. Available: https://cloudsecurityalliance.org/research/secaas/. [Accessed January 2013].

[30] M. Schmidt, L. Baumgartner, P. Graubner, D. Bock and B. Freisleben, "Malware Detection and Kernel Rootkit Prevention in Cloud Computing Environments," in *19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2011.

[31] D. Goodin, "Webhost Hack Wipes Out Data for 100,000 Sites," 8 June 2009. [Online]. Available: http://www.theregister.co.uk/2009/06/08/webhost_attack/. [Accessed 15 January 2013].

[32] V. S. Subashini, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, pp. 1-11, 2011.

[33] R. Bhadauria and S. Sanyal, "Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques," *International Journal of Computer Applications*, vol. 47, no. 18, pp. 47-66, 2012.

[34] C. Modi, D. Patel, B. Borisaniya, A. Patel and M. Rajarajan, "A survey on security issues and solutions at different layers of Cloud computing," *The Journal of Supercomputing*, pp. 1-32, 2012.

[35] H. Zeng, "Research on Developing an Attack and Defense Lab Environment for Cross Site Scripting Education in Higher Vocational Colleges," *Int'l Conf. Computational and Information Sciences,* 2013.

[36] M. S., Siddiqui, D Verma, "Cross site request forgery: A common web application weakness," *IEEE International Conference on Communication Software and Networks* (ICCSN), pp.538-543, 2011.

[37] G. Pék, L. Butty´an, and B. Bencsáth. A survey of security issues in hardware virtualization. *ACM Computer Surveys*. 45, 3, Article 40 (July 2013), 34 pages.

[38] M. Abbasy and B. Shanmugam, "Enabling Data Hiding for Resource Sharing in Cloud Computing Environments Based on DNA Sequences," in *IEEE World Congress*, 2011.

[39] J. Feng, Y. Chen, D. Summerville, W. Ku, Z. Su, "Enhancing cloud storage security against roll-back attacks with a new fair multi-party non-repudiation protocol," in *Consumer Comm. and Networking Conf.*, 2011.

[40] L. Kaufman, "Data security in the world of cloud computing," *IEEE Security and Privacy Journal*, vol. 7, no. 4, pp. 61-64, 2009.

[41] E. Blasch, Y. Chen, G. Chen, D. Shen, and R. Kohler, "Information Fusion in a Cloud-Enabled Environment," in Keesook Han, Baek-Young Choi, Sejun Song (Eds.), *High Performance Cloud Auditing and Applications*, Springer Publishing, 2013.

[42] E. Blasch, G. Seetharaman, and K. Reinhardt, "Dynamic Data Driven Applications System concept for Information Fusion," *Procedia Computer Science*, Vol. 18, pp. 1999-2007, 2013.

[43] B. Liu, E. Blasch, Y. Chen, A. J. Aved, *et al.*, "Information Fusion in a Cloud Computing Era: A Systems-Level Perspective," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 29, No. 10, pp. 16 – 24, 2014.

[44] M. Rosenblum and T. Garfinkel, "When virtual is harder than real: security challenges in virtual machine based computing environments," in *10th conference on Hot Topics in Operating Systems*, Berkeley, 2005.

[45] 16 September 2009. [Accessed January 2013].]. Available:
http://www.cyber.st.dhs.gov/docs/National_Cyber_Leap_Year_Summit_2009_Co-Chairs_Report.pdf.

[46] M. Dunlop, S. Groat, W. Urbanski, R. Marchany and J. Tront, "MT6D:a moving target IPv6 defense," in *IEEE Military Communications Conference (MILCOM),* 2011.

[47] R. Zhuang, S. Zhang, S. A. DeLoach, X. Ou and A. Singhal, "Simulation-based approaches to studying effectiveness of moving-target network defense," in *National Symposium on Moving Target Research*, 2012.

[48] S.Narain, "Moving Target Defense With Configuration Space Randomization," [Accessed 30 Jan. 2013]. Available: https://www.ncsi.com/nsatc11/presentations/thursday/emerging_technologies/narain.pdf.

[49] K. Kim, "ROAFTS: A Middleware Architecture for Real-time Object-oriented Adaptive Fault Tolerance Support," in *IEEE CS 1998 High-Assurance Systems Engineering (HASE) Symp.*, Washington, D.C., 1998.

[50] A. Tyrrell, "Recovery Blocks and Algorithm Based Fault tolerance," in *22nd EUROMICRO Conf.*, 1996.

[51] K. Kim and H. Welch, "Distributed Execution of Recovery Blocks: An Approach for Uniform Treatment of Hardware and Software Faults in Real-Time Applications," *IEEE transactions on Computers*, vol. 38, no. 5, pp. pp. 626-636, 1989.

[52] W. Toy, "Fault-Tolerant Computing," in *Advances in Computers*, Academic Press, pp. 201-279, 1987.

[53] A. Avizienis, "The N-Version Approach to fault Tolerant Software," *IEEE transactions on Software Engineering*, Vols. SE-11, no. 12, 1985.

[54] S. Latif-Shabgahi, "An Integrated Voting Algorithm for Fault Tolerant Systems", in *Proc. International Conference on Software and Computer Applications (IPCSIT*), vol. 9, 2011.

[55] D. Evans, A. Nguyen-Tuong and J. Knight, "Effectiveness of Moving Target Defenses," in *Advances in Information Security*, Springer, 2011, pp. 29-39.

[56] "PaX Homepage," 2000. [Online]. Available: http://pax.grsecurity.net/. [Accessed October 2012].

[57] E. Barrantes, D. Ackley, S. Forrest, *et al.*, "Intrusion Detection: Randomized Instruction Set Emulation to Disrupt Binary Code Injection Attacks," *ACM Conf. on Computer and Communications Security*, 2003.

[58] C. Cadar, P. Akritidis, M. Costa, J.-P. Martin, M. Castro, "Data Randomization," *Microsoft Research*, 2008.

[59] L. Ge, W. Yu, D. Shen, G. Chen, K. Pham, *et al.*, "Toward Effectiveness and Agility of Network Security Situational Awareness using Moving Target Defense (MTD)," *Proc. SPIE*, Vol. 9085, 2014.

[60] P. Verissimo, A. Bessani and M. Pasin, "The TClouds Architecture: Open and Resilient Cloud-of-clouds Computing," *in IEEE/IFIP 42nd Int'l Conf. on Dependable Systems and Networks Workshops*, 2012.

[61] G. Vallee, C. Engelmann, A. Tikotekar, T. Naughton, K. Charoenpornwattana, C. Leangsuksun and S. Scott, "A Framework for Proactive Fault Tolerance," *Int'l Conf. on Availability, Reliability and Security*, 2008.

[62] A. Keromytis, G. R. S. Sethumadhavan, S. Stolfo, Y. Junfeng, A. Benameur, M. Dacier, M. Elder, D. Kienzle and A. Stavrou, "The MEERKATS Cloud Security Architecture," in *32nd International Conference on Distributed Computing Systems Workshops*, 2012.

[63] D. Luo and J. Wang, "CC-VIT: Virtualization Intrusion Tolerance Based on Cloud Computing," in 2nd *International Conference on Information Engineering and Computer Science*, 2010.

[64] D. Jeffrey, G. Sanjay, "MapReduce: Simplified Data Processing on Large Clusters," In *Communications of the ACM*, 2008.

[65] E. Cheng, L. Ma, A. Blaisse, *et al.*, "Efficient Feature Extraction from Wide Area Motion Imagery by MapReduce in Hadoop," *Proc. SPIE*, Vol. 9089, 2014.

[66] L. Ge, H. Zhang, G. Xu, W. Yu, *et al*, "Towards MapReduce Based Machine Learning Techniques for Processing Massive Network Threat Monitoring Data," *Networking for Big Data*, CRC Press, Taylor & Francis Group, 2015.

[67] E. Blasch, "Level 5 (User Refinement) issues supporting Information Fusion Management" *Int. Conf. on Info Fusion*, July 2006.

[68] E. P. Blasch, E. Bosse, and D. A. Lambert, *High-Level Information Fusion Management and Systems Design*, Artech House, Norwood, MA, 2012.

[69] B. Liu, Y. Chen, A. Hadiks, *et al.*, "Information Fusion in a Cloud Computing Era: A Systems-Level Perspective," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 29, No. 10, pp. 16 – 24, Oct. 2014.

[70] L. Peng, D. Lipinski, K. Mohseni, "Dynamic Data Driven Application System for plume estimation using UAVs," *Journal of Intelligent and Robotic Systems*, Vol. 74, pp. 421-436, 2014

[71] E. Blasch, "Context aided Sensor and Human-based Information Fusion" *IEEE National Aerospace and Electronics (NAECON)*, 2014.

[72] E. Blasch, J. Nagy, A. Aved, W. M. Pottenger, M. Schneider, R. Hammoud, E. K. Jones, A. Basharat, *et al.*, "Context aided Video-to-Text Information Fusion," *Int'l.. Conf. on Information Fusion*, 2014.

[73] W. Li, Y. Badr, F. Biennier, "Service Farming: An Ad-hoc and QoS-aware Web Service Composition Approach," *Dans SAC 2013*, Coimbra, Portugal, pp. 750-756, 2013.

[74] R. M. Dijkman, M. Dumas, and C. Ouyang. "Semantics and analysis of business process models in BPMN," *Information and Software Technology*, 50 (12), 1281-1294, 2008.

[75] C. Tunc, F. Fargo, Y. Al-Nashif, S. Hariri, J. Hughes, Autonomic Resilient Cloud Management (ARCM), *ACM Int. Conference on Cloud and Autonomic Computing*, (CAC '14), 2014.

[76] G. Dsouza, G. Rodríguez, Y. B. Al-Nashif, S. Hariri, "Building resilient cloud services using DDDAS and moving target defence," *IJCC* 2(2/3): 171-190, 2013.

[77] E. Blasch, Y Al-Nashif, and S. Hariri, "Static versus Dynamic Data Information Fusion analysis using DDDAS for Cyber Trust," *International Conf. on Computational Science*, Procedia Computer Science, 2014.

[78] A-M. Barthe-Delanoë, *et al*. "A platform for event-driven agility of processes: A delivery context use-case." *Collaborative Systems for Reindustrialization*. Springer Berlin Heidelberg, 2013. 681-690.

[79] E. Blasch, R. Breton, and P. Valin, "Using the C-OODA Model for CIMIC Analysis," *Proc. IEEE Nat. Aerospace Electronics Conf (NAECON)*, 2011.

[80] E. P. Blasch, D. A. Lambert, P. Valin, *et al.*, "High Level Information Fusion (HLIF) Survey of Models, Issues, and Grand Challenges," *IEEE Aerospace and Electronic Systems Mag.,* Vol. 27, No. 9, Sept. 2012.

[81] E. Blasch, A. Steinberg, S. Das, J. Llinas, C.-Y. Chong, O. Kessler, E. Waltz, and F. White, "Revisiting the JDL model for information Exploitation," *Int'l Conf. on Info Fusion*, 2013.

[82] E Blasch, J Dezert, B Pannetier, "Overview of Dempster-Shafer and belief function tracking methods," *Proc. SPIE*, Vol. 8745, 2013.

[83] P. C. G. Costa, K. B. Laskey, *et al.*, "Towards Unbiased Evaluation of Uncertainty Reasoning: The URREF Ontology," *Int. Conf. on Info Fusion*, 2012.

[84] E. Blasch, K. B. Laskey, A-L. Joussselme, V. Dragos, P. C. G. Costa, and J. Dezert, "URREF Reliability versus Credibility in Information Fusion (STANAG 2511)," *Int'l Conf. on Info Fusion*, 2013.

[85] E. Blasch, A. Jøsang, J. Dezert, P. C. G. Costa, K. B. Laskey, A-L. Jousselme, "URREF Self-Confidence in Information Fusion Trust," *Int'l. Conf. on Information Fusion*, 2014.

[86] E. Blasch, "Trust Metrics in Information Fusion," *Proc. SPIE*, Vol. 9091, 2014.

[87] A. N. Steinberg, C. L. Bowman, G. Haith, *et al.*, "Adaptive Context Assessment and Context Management," *Int'l.. Conf. on Information Fusion*, 2014.