Dhinaharan Nagamalai
Jan Zizka (Eds)

Computer Science & Information Technology

5th International Conference on Computer Science and Engineering
(CSEN-2018) August 25 ~ 26, 2018, Dubai, UAE

AIRCC Publishing Corporation

Volume Editors

Dhinaharan Nagamalai,
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

Jan Zizka,
Mendel University in Brno, Czech Republic
E-mail: zizka.jan@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-921987-90-8
DOI : 10.5121/csit.2018.81201 - 10.5121/csit.2018.81204

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 5th International Conference on Computer Science and Engineering (CSEN-2018) was held in Dubai, UAE during August 25 ~ 26, 2018. The 4th International Conference of Networks, Communications, Wireless and Mobile Computing (NCWC 2018) was collocated with The 5th International Conference on Computer Science and Engineering (CSEN-2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CSEN-2018, NCWC-2018 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, CSEN-2018, NCWC-2018 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CSEN-2018, NCWC-2018.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

Dhinaharan Nagamalai
Jan Zizka

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Abdelhalim BOUTARFA	University of Batna , Algeria
Abderrahmane Nitaj	University of Caen Normandie, France
Aberto Magrenan	International University of La Rioja (UNITE), Spain
Agoujil Said	University of Moulay Ismail Meknes, Morocco
Ahmad Qawasmeh	The Hashemite University, Jordan
Ahmad T. Al-Taani	Yarmouk University, Jordan
Ahmed Mohamed Khedr	Sharjah University, UAE
Ali Javadi	Iran University of Science and Technology, Iran
Ali Salem	University of Sfax,Tunisia
Ameera Saleh. Jaradat	Yarmouk University, Jordan
Amel B.H.Adamou-Mitiche	University of Djelfa, Algeria
Amizah Malip	University of Malaya, Malaysia
Annamalai	Prairie View A&M University, USA
Ashok Kumar T.A.	Garden City University, India
Basar Oztaysi	Istanbul Technical University, Turkey
Bing Zhou	Sam Houston State University, USA
Bouchra Marzak	Faculty of Sciences - Hassan II University, Morocco
Carmen Martinez	University of Jaen, Spain
Chang-Hyun	Korea Marine Equipment Research Institute, Korea
Chin-Chih Chang	Chung-Hua University, Taiwan
Christophe NICOLLE	University of Bourgogne Franche, France
Da Yan	The University of Alabama at Birmingham, USA
Dabin Ding	University of Central Missouri, USA
Debabrata Datta	St. Xavier's College, India
Farhad pourfarzi	Ardabil University of Medical Sciences, Iran
Figen Balo	Firat University, Turkey
Florence SEDES	Toulouse University, France
Guoqing Xiao	Hunan University, China
Hadi Amirpour	Universidade da Beira Interior, Portugal
Haibat Jadhav	Flora Institute of Technology, India
Haibo Yi	Shenzhen Polytechnic, China
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hamzeh Khalili	Universitat Politecnica de Catalunya (UPC), Spain
Hani Bani-Salameh	Hashemite University, Jordan
Hasnaoui Salem	University Tunis El-Manar, Tunisia
Houda KHROUF	Atos Innovation Lab, France
Hung Tran Cong	Posts and Telecoms Institute of Technology, Viet Nam

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

4th International Conference on Networks, Communications, Wireless and Mobile Computing (NCWC 2018)

5th International Conference on Computer Science and Engineering (CSEN-2018)

DATA PROVENANCE IN THE INTERNET OF THINGS: VIEWS AND CHALLENGES

Zuhaib Bari Mufti[1] and Mahmoud Elkhodr[2]

[1]School of Computing, Engineering, and Mathematics, Western Sydney University, Sydney, Australia
[2]School of Engineering and Technology, Central Queensland University, Sydney, Australia

ABSTRACT

The IoT is a rich and dynamic network of interconnected networks where various devices share information, create knowledge and perform actuations events. In such an environment, it is important to precisely trace the origin of data and the events that contributed to their changes. This concept has long been known as provenance. This paper attempts to shade some lights on the importance of data provenance in the IoT, its application, and the challenges associated with data provenance in the IoT.

KEYWORDS

Internet of Things, Wireless Network, Data Provenance, Security & Trust

1. INTRODUCTION

Earlier forms of provenance appeared as a method to validate the authenticity of an artefact by examining an object's origin, ownership or any modifications made to the item (Dogan, 2016a). In a world entangled in a mesh of connected networks i.e. the Internet of Things (IoT), provenance becomes even more vital to keep track of events, the source of information, decisions, and origin of data and the metadata. E-Science relies on provenance to measure the quality of the data[1]. Nowadays, data provenance is no longer just concerned with finding the origin of the data, but it extends to include the capacity of tracking any events or modification made to the data. Example includes the followings applications[2]:

- Creating a file and any subsequent modifications to it and defining the ownership and accessibility is a form of File Systems provenance[1].
- Administrative systems and intrusion detection aided by logging system events is a form of Operating systems provenance.
- Similarly, compliers and run time errors can be detected by tagging the source line using compilers.
- Records of any insertion, modification and deletion are an application of provenance in curated databases[2].
- Browsing history is considered a form of web browsing provenance.

Additionally, several financial institutions are required by laws to record the source and origin of each digital transaction. This highlights the importance of provenance in the financial industry where each paper notes and its origin is treated as provenance. Intelligence and hospital systems are some of the prime users of provenance information[2]. A discrete information system having

Dhinaharan Nagamalai et al. (Eds) : CSEN, NCWC - 2018
pp. 01–07, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81201

adequate relevance, capable of undergoing classification into various domains for the purpose of evaluation can be considered as Intelligence. Hospital records and related data protected by the Health Care Portability and Accountability Act (HIPAA) act makes it obligatory to record and store all hospital records and data in addition to managing proper authorised access to the data[3].Information and lineage data used as provenance must possess some inherent technical features in order for it to be reliable. Some of these features are as follows:

- Information about every action performed on data needs to be preserved and stored completely[4].
- Ensuring that no manipulation of the data with a malicious intent takes place (Integrity).
- Provenance data should be available readily without any hassle (Availability).
- By providing authorised access to provenance data, confidentiality of the information can be ensured[5].
- Provenance data in the E-science field must be obtained in an economically feasible manner.
- Provenance data must be stored and available in such a way that the privacy of a person is not compromised, especially in the IoT[6]. Systems involving data provenance data need to deal with diverging aspects of ensuring that no outside entity or system is able to access the data and at the same time data within the system is readily available and shared among authorised entities for transparency[5].

2. APPLICATIONS OF DATA PROVENANCE

Some of the most common applications of provenance have been listed below:

DIAGNOSTICS:

Provenance has been used for debugging and detecting real time anomalies in a distributed system [7] If a monitoring system is based on declarative monitoring, there is a provision to analyse the network traffic which indirectly can be employed for detecting an intrusion[8]. SeNDlog can dynamically trace changes to a routing table and helps in generation of an alarm if the number of changes made are above a certain threshold value. Once an alarm has been generated a distributed recursive query on the network performance can trace the origin of any malicious activity[9].

SECURITY:

Data provenance covers historical data in addition to real time data as well. This helps in finding correlations in the network pattern of an attacker; thus, helping in the security of vital assets. Locating the source or filtering the IP address from the traffic is a typical example[10]. Annotations can be used in data provenance to help identifying potential attacker as well as tracing back information for forensic analysis[8]. Provenance can also be used to identify any malicious packets dropping in a sensor network[11].

ACCOUNTABILITY:

Data Provenance ensures a proper accountability for an action as well as data. In conventional forensic analysis, call-details consisting of information, time and location of the call are a form of data provenance. Network Provenance can be also used to manage trusts in a distributed environment[12].

TRUST:

By enabling a network of information where nodes are capable of tracing the origin of data, effective trust policies can be implemented[8]. Multi-hop networks and Body Sensor Networks rely also on data provenance to ensure trust[13]. Provenance can be used in quantifying trust, which enables sensors to process information from trusted nodes only (Wenchao et al., 2008).

OPTIMIZATION:

Monitoring of a system and tracing important events using data provenance in sensor networks can help in optimization of resources[2]. Resource allocation and finding bad routes or draining nodes are good examples.

DEVELOPMENT PROCESS:

Provenance logs can be used to capture changes in a network before and after an event takes place[14]. Comparing the snapshots before and after to see the changes in energy and other resources and using provenance to gauge the dependencies of a system can help in the development of a smooth process.

RECOVERY:

Provenance is often used to restore a system after a failure and for success validation[2]. In a sensor network, it is vital to not only identify the points of failure but also to avoid those which cause system anomalies. Provenance of graphs plays a key role in scenarios requiring troubleshooting as well.

3. DATA PROVENANCE CHALLENGES IN THE IOT

The IoT proposes various revolutionary concepts by employing millions, even billions, of tiny sensor or actuators nodes collecting and communicating information just about everything[15]. The volume of data collected in such a large network will have a high velocity, volume and divergent variety. This augments the significance of analysing the data for trustworthiness establishment in order to make better decisions. Therefore, it is becoming increasingly important to analyse a distributed network for possible anomalies and to pinpoint any erring node. These capabilities are some of the functional requirements needed to provision for Network Accountability and forensic analysis. Therefore, provenance of information or data plays a critical role in such environments. On the other hand, in an IoT smart based environment, the flow of information is relayed ultimately through the open Internet. It is a well-known security principle that the Internet is insecure. Therefore, it is essential to have reliability, trust, accountability and similar security principles addressed by employing a strong provenance enabled system.

To this end, as new, comples and dynamic data exchanged by IoT devices gets published on the Internet -where platforms accessing, publishing and modifying the data can be also diverse-, it becomes important to address the lineage, trustworthiness, reliability and accuracy of data in the IoT[16]. While papers' provenance has been employed in several systems, the IoT poses some unique challenges to the provisioning of data. Some of the challenges are listed below.

SECURITY

Data transmitted through an IoT system is extremely susceptible to attacks by a third party[17]. If provenance of data is insecure, it can result in a breach of sensitive information. The challenge is to impart enough confidentiality so that provenance can be accessed by only authorised individuals. Under certain circumstances, identity and location of the IoT device needs to be secured above all as the device may be more valuable than the data it sends. A robust security mechanism should incorporate confidentiality, integrity, privacy and availability of the information[18]. However, a high level of heterogeneity coupled with the massive scale in which IoT devices are likely to be deployed complicates the security issue of data provenance in the IoT[18].Moreover, IoT devices lack the computational power and energy requirements to incorporate complex security solutions such as encryption, cryptography, public key and symmetric key infrastructure[19]. Integrity of data provenance to assure a level of trust should be considered as well. This demands the use of cryptographic hashes algorithms which are extremely difficult to implement in the IoT due to the resource constraint feature of IoT devices[17].

BIG DATA

The massive volume of data produced by sensor networks in the IoT can result in the generation of petabytes of data, thus resulting in additional computational burden on the already fragile system[20]. Some researchers point out to the fact that Big Data and IoT need to be treated in tandem rather than as separate entities[21]. Querying and tracing Provenance information in such a system to point out the anomalies and other faults in the system is extremely difficult. Data Provenance may consume a lot of network resources, which in turn may hamper the operational efficiency of the system[17]. To ensure that Metadata is readily available upon request, there is a need to design systems which have a very low computational overhead to ensure smooth performance[22].

INDEXING:

A complete list of provenance in an IoT environment is practically impossible owing to the large nature of information. Hence, an indexing scheme is normally used[22]. However, it is likely that information can't be queried in a conventional manner wherein looking-up an attribute to retrieve the data is common. Users often have to query the dataset, which is essentially a subset of an attribute. Even in XML-based schema used for mapping names and values may prove not to be sufficient without the help of additional structures.

MULTIPLE CONSUMERS:

IoT data can have potentially vast and diverse range of consumers, with clients possessing divergent requirements. Some clients may need data on a real time bases, whilst others may just need to archive the provenance data. For example, while managing a smart city environment, provenance data may be required dynamically to make better decisions and rectify any anomalies in a system. Therefore, adequate flexibility is required for the provenance of data in the IoT.

TRANSFORMATION OF DATA:

Sensors in an IoT network collect data and pass or route them to other sensors, which may modify the information before passing it on to a more computationally powerful device. In other cases, actuators may receive data modified by various sensors during the transfer phase and thus, it becomes necessary to overcome the challenges encountered in representing such a complex provenance of information[17].

QUERYING INFORMATION:

Just tracing the lineage of data and its object may not be adequate for future systems and powerful querying tools need to be deployed to meet the cybersecurity challenges of next generation [23]. Records may need to be queried based on the context and requirements while maintaining the confidentiality at the same time may be essential [17].

INTEROPERABILITY:

IoT devices need to work in an extremely interoperable environment to ensure that the data collected by the sensors is successfully delivered to the target location. Also, various intermediate nodes or platforms are capable of reading or modifying the data. In such as case, Data Provenance demands that all the devices present in a system to be interoperable by having sufficient features to use each other's data. Keeping in view the limited computational power and resources of IoT devices and ensuring security of the system, achieving efficient interoperability in the IoT is still not an easy task[24]. IoT devices are manufactured by different vendors and may use different networking and routing protocols and often there is no standard or regulation yet in place to ensure uniformity and interoperability of devices.

DATABASE MANAGEMENT:

IoT data can be discrete, continuous, and dynamic. Certain data can be descriptive or based on environmental factors. Other can be in the form of addresses such as RFID tag format[25]. As the number of IoT devices may run into Billion coupled with limited computational capability of devices, it is almost impossible to adhere to IPv4 protocol for IoT Devices. Thus Internet Engineering Task Force (IETF) has introduced various protocols for IoT based in IPv6 addressing format[26]. But in doing so the header size has been increased from 32 bit to 128 bit addressing scheme, thus making it extremely difficult for resource constrained IoT devices to implement the system[25]. Thus, traditional databases may not provide a complete solution for such a complex system and it becomes imperative to deploy innovative and non-traditional databases.

An innovative approach is needed to cope up with the challenges associated with data provenance in IoT. In this case numerous protocols have been put forward such as the 6LoWPAN (IPv6 over low power wireless personal area network) protocol which is specially designed for resource constrained devices. The protocol is based on IPv6 and ensures universality, stability and additional features for IoT devices[26]. 6LoWPAN protocol suite specifically targets the integration of IPv6 and MAC (Media Access Control) and physical layers used in IEEE 802.15.4 standard. It is pertinent to mention that the maximum frame size of 127 bytes supported by IEEE 802.15.4 standard hinders the use of IPv6 and MAC header. By incorporating such a technology, it is possible to address various security and provenance issues using symmetric key and public key cryptography solutions.

One must also considers that not all IoT devices can transmit data. Hence, IoT gateways are used in some cases to bridge between the IoT devices with the Internet. Therefore, helping in harnessing the full potential of the technology[27]. The gateways provide a mechanism to ensure the computational power of IoT devices does not need to be high enough to increase the overall cost of the system, but at the same time they are able to smoothly operate in tandem with external applications and computational devices without compromising the efficiency and effectiveness of the system. Constricted application Protocol (CoAP) for device to device communication is employed to enables IoT devices to use the Representational state transfer (REST) mechanism which is similar to HTTP. This enables data provenance to be written using standard HTTP queries, which helps in mitigating the complexities of collecting provenance of data in IoT

applications. The use of several NoSQL like CouchDB, MongoDB etc. databases to store provenance data is recommended as they enable extensive flexibility during storing and retrieving of information.

4. CONCLUSIONS

The IoT with its diverse and heterogeneous nature of communications requires the provisioning of data provenance. Undoubtedly challenges associated with data provenance, especially in the IoT are enormous owing to the constrained resources available to IoT devices. Certain areas such as in the health and security domains demand elaborated provenance mechanisms whereas such intricacies may not be desired in simple IoT application such as controlling lighting in a smart building. Our future work will look into solutions that employs a middleware to leverage the overhead associated with the provision of data provenance in the IoT.

REFERENCES

[1] J. Cheney, S. Chong, N. Foster, M. Seltzer, and S. Vansummeren, "Provenance: a future history," in Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, 2009, pp. 957-964: ACM.

[2] G. Dogan, "A Survey of Provenance in Wireless Sensor Networks," Adhoc & Sensor Wireless Networks, vol. 31, 2016.

[3] R. Lange, "Provenance aware sensor networks for real-time data analysis," University of Twente, 2010.

[4] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," in FAST, 2009, vol. 9, pp. 1-14.

[5] S. Bauer and D. Schreckling, "Data provenance in the internet of things," in EU Project COMPOSE, Conference Seminar, 2013.

[6] D. Gollmann, "Computer security," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 5, pp. 544-554, 2010.

[7] A. Singh, P. Maniatis, T. Roscoe, and P. Druschel, "Distributed monitoring and forensics in overlay networks," 2006: EuroSys.

[8] W. Zhu, E. Cronin, and B. T. Loo, "Provenance-aware secure networks," Departmental Papers (CIS), p. 387, 2008.

[9] W. Zhou, E. Cronin, and B. T. Loo, "Provenance-aware declarative secure networks," 2007.

[10] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," IEEE/ACM transactions on networking, vol. 9, no. 3, pp. 226-237, 2001.

[11] S. Sultana, E. Bertino, and M. Shehab, "A provenance based mechanism to identify malicious packet dropping adversaries in sensor networks," in Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on, 2011, pp. 332-338: IEEE.

[12] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The role of trust management in distributed systems security," in Secure Internet Programming: Springer, 1999, pp. 185-210.

[13] K. Govindan et al., "Pronet: Network trust assessment based on incomplete provenance," in MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011, 2011, pp. 1213-1218: IEEE.

[14] K.-K. Muniswamy-Reddy, Foundations for provenance-aware systems. Harvard University Cambridge, 2010.

[15] H.-S. Lim, Y.-S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in Proceedings of the Seventh International Workshop on Data Management for Sensor Networks, 2010, pp. 2-7: ACM.

[16] O. Hartig, "Provenance Information in the Web of Data," LDOW, vol. 538, 2009.

[17] A. Alkhalil and R. A. Ramadan, "IoT Data Provenance Implementation Challenges," Procedia Computer Science, vol. 109, pp. 1134-1139, 2017.

[18] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Computer networks, vol. 76, pp. 146-164, 2015.

[19] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431-440, 2015.

[20] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the internet of things era," IT Professional, vol. 17, no. 3, pp. 32-39, 2015.

[21] S. Sahu and Y. Dhote, "A Study on Big Data: Issues, Challenges and Applications," International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), vol. 4, no. 6, pp. 10611-10616, 2016.

[22] A. Chebotko, J. Abraham, P. Brazier, A. Piazza, A. Kashlev, and S. Lu, "Storing, indexing and querying large provenance data sets as RDF graphs in apache HBase," in Services (SERVICES), 2013 IEEE Ninth World Congress on, 2013, pp. 1-8: IEEE.

[23] V. Gazis et al., "Short paper: IoT: Challenges, projects, architectures," in Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on, 2015, pp. 145-147: IEEE.

[24] P. Grace, J. Barbosa, B. Pickering, and M. Surridge, "Taming the interoperability challenges of complex iot systems," in Proceedings of the 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT, 2014, pp. 1-6: ACM.

[25] J. Cooper and A. James, "Challenges for Database Management in the Internet of Things," IETE Technical Review, vol. 26, no. 5, pp. 320-329, 2009/09/01 2009.

[26] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities," IEEE Wireless Communications, vol. 20, no. 6, pp. 91-98, 2013.

[27] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway," ICT Express, 2017/04/27/ 2017.

INTENTIONAL BLANK

SECURITY AND PRIVACY IN IOT ENVIRONMENT: A SYSTEMATIC MAPPING STUDY

Moussa WITTI and Dimitri KONSTANTAS

Information Science Institute University of Geneva
Route de Drize 7, 1227 Carouge, Switzerland

ABSTRACT

Over the last decades, the rapid grow of Internet of Things or IoT connected to the Internet has accelerated sensitive and non-sensitive data exchange such as lifestyle, personal data (using we arables sensors, smart devices). A huge number of heterogeneous sensors may convey or collect and dispatch sensitive data from an endpoint to worldwide network on Internet. Privacy remain an important issues, therefore Internet of Things has developed significant attention in the research. In this paper, we aim to evaluate current research state related to privacy and security in IOT by identifying existing approaches and publications trends. Therefore, we have conducted a systematic mapping study using auto mated searches from selected relevant academics databases. The result of this mapping highlights research type and contribution in different facets and research activities trends in the topic of "security and privacy" in IoT edge, cloud and fog environment.

KEYWORDS

Internet of Thing, privacy, security, mapping study

1. INTRODUCTION

Recently we are witnessing the increase use of "Internet of Things" (IoT). According to Gartner's report on "IoT Technology Disruptions", IoT security market will grow from $547 million in 2018 to $841 million by the end of 2020. Gartner predicts that the use of the IoT will increase of 31% up. Approximately 67% of the use of IoT will be located in North America, Western Europe and China. From RFID technologies in supply chain management, to wearables devices in lifestyle or healthcare monitoring system, and smart sensors in automotive or in home automation, the use of "Internet of things" has led to change our life.

However, data collection raises privacy and security issues in Internet of Things (IoT) environment. Using heterogeneous protocols that are WiFi, Bluetooth, ZigBee, sub-GHz, Z- Wave, Thread and 2G/3G/4G cellular, along end-to-end communication how to ensure security and preserve privacy?

In this paper, we conducted a systematic mapping study to perform thematic analysis, trends and future works about security and privacy-preserving methods and models in IOT environment.

Dhinaharan Nagamalai et al. (Eds) : CSEN, NCWC - 2018
pp. 09–19, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81202

The paper is organized as follows: Section 2 presents related work in the research, Section 3 defines the research method, Section 4 provides the results of the systematic mapping and describes overview of included studies while Section 5 try to respond to the research questions and discussing main findings. Section 6, deal with the threats to research validity and Section 7 provides conclusions and directions for future work.

2. RELATED WORK

In the literature, privacy and security issues are challenged and several security models for IoT have been designed. The rapid growth of IoT has extended Internet to any small smart devices in distributed environment [11] thus has introduced a serious problematic. As IoT environment is more heterogeneous, more complex [17] and maintaining security is very critical in distributed system as well as cloud and fog environment [1] [2] [23] [32].

Most research studies [5] [8] [16] [17] [18] [27] are focused on how to integrate security among application, perception and transport layers level for distributed or cloud environment such as IaaS (Infrastructure as a Service), SaaS (Software as Service), and PaaS (Platform as Service). Except rare studies [12] [25] [33], focused on specific use, we found only one research paper using systematic mapping study on IOT and cloud computing [7].

To protect sensitive data a huge of privacy-preserving algorithms have been developed such as k-anonymity, l-diversity. The concept of k-anonymity has been introduced by L. Sweeney and P.Samarati [21] in order to preserve privacy. While l-diversity is a data anonymization technique based on generalization and suppression often with a loss of the quality of the information. L-diversity is defined as extension of the k-anonymity [30]. Another algorithm "t- closeness" [22] has been developed to anonymize data [8][20]. This technique is an extension of l-diversity and designed to preserve the confidentiality of sensitive data while reducing the granularity of data representation.

3. RESEARCH METHOD

In the experimental software engineering, there are two main approaches to conduct a literature reviews that are “Systematic Mapping Studies” and “Systematic Literature Reviews”. If a researcher aims to identify, classify, and evaluate result to respond for a specific research question “Systematic Literature Reviews” is the adequate approach but if he seeks to answer for multiples research questions “Systematic Mapping Study” is the best one. In this paper, we have conducted the formal guidelines of Systematic Mapping Study from Petersen et al. [25] performed in five steps. The outcome from each step gives the input for the next step. SMS start with the initial research questions built up to provide a general scope for the study used to find out research papers (step 2) from the selected digital libraries (according the research fields). In the next step, screening process start with a set of inclusion and exclusion criteria to select relevant papers (step 3). Finally, the keywording process (step 4) enable classification and data extraction (step 5) wich would have to answer the research questions (figure 1).

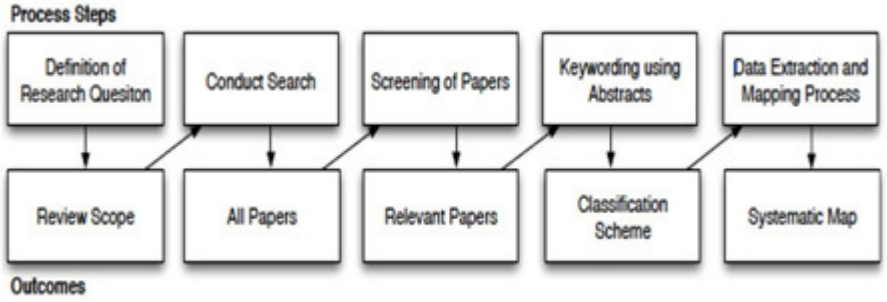

Figure 1: Systematic Mapping Process

3.1 OBJECTIVES

We aim to assess how privacy and security are managed in different context of use of IOT such as IOT edge, IOT cloud and fog environment. We aim to analyse and map research studies that deal with security and privacy concerns in the use of Internet of Thing. Thus, we conducted systematic mapping study to assess data securisation methods and privacy-preserving technique in the research field. We aim also to analyze research contributions and trends in IOT edge, IOT cloud and fog environment.

3.2 RESEARCH QUESTIONS

"How security and privacy are provided and maintained in IOT enabled-technologies?"

We have decomposed this main research question into four specific questions as show in the table 1.

Table 1. Research questions

N°	Questions	Motivation
RQ1	What are security and privacy issues in IOT environment?	Find out an overview of studies about security and privacy in IOT.
RQ2	What are the field of the studies?	Find out the context of the related studies.
RQ3	How provide security and privacy in IOT environment?	Explore the state of the related research activities and their evolution.
RQ4	What are research trends in IOT environment about security and privacy concerns?	Assess future trend of the research activities about security and privacy in IOT.

3.3 SEARCH STRATEGY

According to our research questions, we have built up our Search Strings formulated using general terms with **AND** clause as **"IOT AND (security and privacy)"**. We adopted use of boolean operator such as **"AND"** to focus our search only on specific subjects. Therefore, we have restricted our search items to select from digital libraries only scientific papers, with as the specified keywords related to security and privacy in IOT. Then we used that search string above in the major search engines for academic studies, which are **ACM**, **DBLP**, **Google Scholar**, **iEEE**, **Science Direct**, **Springer**. For each digital libraries, according to their search rules, we did

some customization in order to adapt our generic search string.

3.4 INCLUSION-EXCLUSION CRITERIA

We filtered out the result of the automatic search; in the relevant academic libraries as explain in the previous subsection, by applying inclusion and exclusion criteria detailed in table 2. The main selection criteria are:

- Selection by **"title"** and by **"abstract"**: we first selected papers only with **"IOT"** and **"privacy"** or **"security"** terms in the title or in the **"abstract"**.

- Selection by full paper reading: we selected paper in English and with a research contribution in IOT environment related to security or privacy.

Table 2. Inclusion / Exclusion Criteria

Inclusion	Exclusion
Studies with title related to IOT.	Paper in other language than English.
Studies related to security in IOT.	Paper without Abstract.
Studies related to privacy concern in IOT.	Paper from workshop.
Studies presenting security and privacy concerns in IOT environments	Books.
Studies about security and privacy in IOT cloud environment.	Studies about other issues in IOT environment.
Studies about security and privacy in IOT fog environment.	Paper out of our scope

4. EXECUTION

The search is executed using automated search engines. During screening process of relevant studies according to our inclusion-exclusion criteria defines previously, we examined firstly title, then abstract and keyword. For those without sufficient details in this part, we did full reading of the content of paper.

4.1 CONDUCTING THE SEARCH

We adapted the search string to each databases and obtained relevant studies in four steps as shown in figure 2:

Step 1: We obtained 3205 studies by putting our search string into the search engines of ACM,DBLP, Google Scholar, iEEE, Science Direct, Springer databases.

Step 2: We remove duplicated studies from more than one source and we obtained 2807 papers.

Step 3: We obtained 522 potentially relevant studies after removing all studies that not matching with research questions.

Step 4: Finally, after applying selection criteria, we obtained 54 relevant papers.

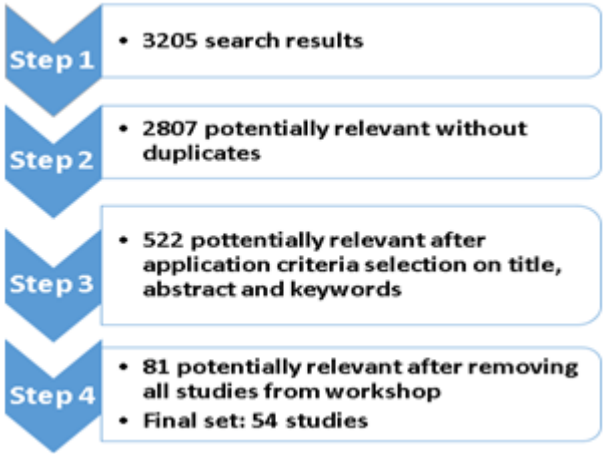

Figure 2: Main steps for selecting relevant studies

4.2 ANALYSIS AND CLASSIFICATION

Keywording phase remains important in the systematic mapping process: after reading firstly title, then abstract and when it is necessary we did full paper reading to search terms and concepts reflecting the research contribution. During abstract or full paper reading process, we categorized relevant papers into one facet or research contribution. Selected studies may be mapped according to their research focus and the context. Then, we can build-up a cluster from classification scheme.
We regrouped all relevant publications into three contributions research type facets such as:

- **Security facet**: IAM (or identity and access management), AAA (or Authentication Authorization Accounting), privacy, K.E.M or (Key Exchange and Management) , trust, confidentiality, integrity, cryptography, availability, I.A.A (or Identification Authentication and Authorization), Anonymity

- Application context or application field facet which are: medical, industrial, public

- Environment facet: IOT Edge, IOT Distributed, IOT cloud, IOT Fog.

In addition, we have obtained flowing research type such as:

- **Opinion papers** : the author gives his views about technical solutions or approach given by others.

- **Survey papers**: In a survey paper, data and results are taken from other papers, the authors draw out some new conclusion.

- Solution proposal

- **Evaluation research papers**

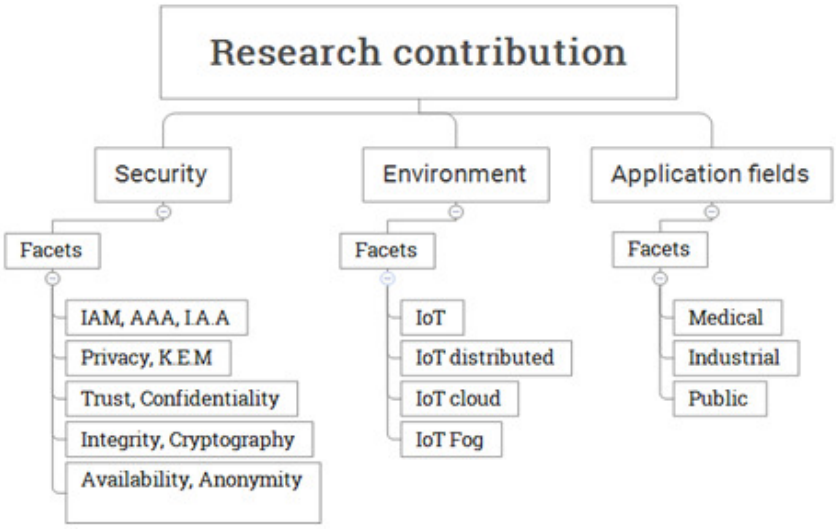

Figure 3: Contributions of relevant studies

5. DISCUSSION

5.1 RQ1: WHAT ARE SECURITY AND PRIVACY ISSUES IN IOT ENVIRONMENT?

Throughout the systematic process, we have identified that all relevant papers discuss about how to secure end-to-end communication. Main issues related to privacy and security are using authentication, data encryption, key exchange mechanisms. Privacy issues are well discussed in general but solutions are not given in details.

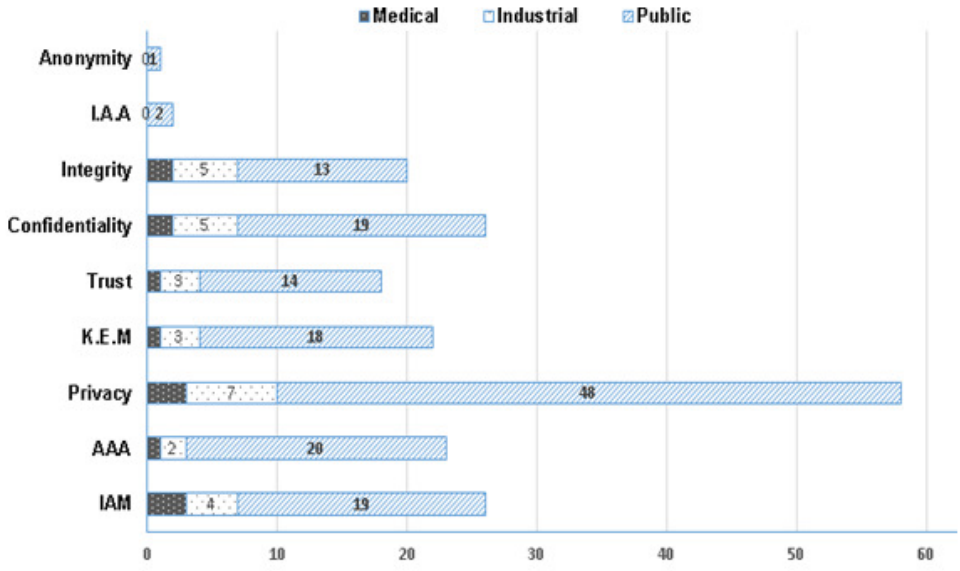

Figure 4: Contributions of relevant studies

5.2 RQ2: WHAT ARE THE FIELD OF THE STUDIES?

Most of the papers (85%) are addressed to public domain. The rest is related to medical (2%) and industrial uses (9%). There are some papers (4%) dealing with all three domains which are public, medical and industrial.

5.3 RQ3: HOW EVOLVED SECURITY AND PRIVACY IN IOT ENVIRONMENT?

From classification scheme in figure 3, the contribution of relevant studies are about mainly authentication, authorization, data encryptions. In the cloud as well as in the distributed IOT environment, the existing securization methods in the literature are used. Most of the selected papers deal with privacy issues without developing algorithms or giving out a methodology.

5.4 RQ4: WHAT ARE RESEARCH TRENDS IN IOT ENVIRONMENT ABOUT SECURITY AND PRIVACY CONCERNS?

From selected papers after data extraction and contributions mapping (figure. 5), we can assess research trends. IOT uses are widespread in public area while in medical and industrial fields IOT remains less developed. On the other hand, most of relevant papers are dealing with security concerns in distributed environment. Globally, privacy concerns in personal sensitive data collection in IOT cloud or fog environment are not detailed and remain in embryonic stage.

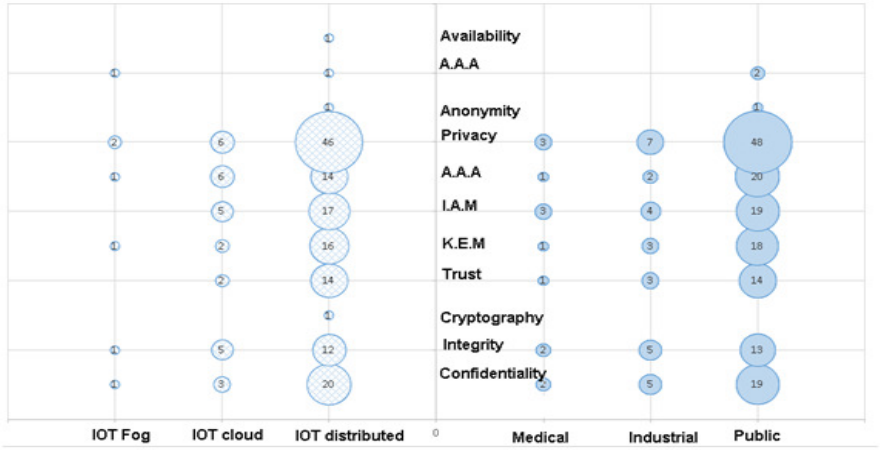

Figure 5: Result mapping of selected studies

6. THREATS TO VALIDITY

In this paper, the method adopted may occur a number of known threats to validity that can bias the result. In our study, we given attention to these threats and more efforts have been done to mitigate these risks. In this section, we list the main threats, which may occur while conducting a Systematic Mapping Study. First, the search criteria of our study were defined on the assumption that our work should only be oriented towards publications related to our research questions (e.g table 1). However, there is a risk that the search engines may have used some relevant publications. To minimize the risk we used unambiguous terms with logical operator to construct a search string. Then, to conduct this study, we selected some academic databases that we considered relevant to our study.

All these digital libraries have been selected on the one hand for their field of application relating to our study and on the other hand for their ranking in scientific research. It is possible that we have not integrated some libraries with relevant publications related to our study. Thus, some relevant articles would be omitted. However, this risk is mitigated by the fact that most of the databases contain a large number of identical items, and we have already experimented with this redundancy in the databases we have selected. Therefore, article redundancy mitigates this risk.

The inclusion and exclusion criteria were defined in a top-down approach from title, abstract, and then full content. First, we have selected publications with title and abstract in English. Then we selected potentially relevant papers that having key terms of our search in the title a nd in the abstract. Finally, we filtered all papers by full content reading according. We probably omit some relevant papers in others languages but our strategy is to execute search string similarly in all selected digital libraries. Finally, the classification scheme of research type or research contribution may be different from one research to another. In our case, we adopted to classify all relevant publications according to the similar terms redundant in their keyword or in their main content.

7. CONCLUSION

By identifying, analysing, classifying publications, we have conducted a systematic mapping study to perform thematic analysis, trends and future works about security and privacy in IOT environment. We have screened 3205 publications, only 54 studies were considered as relevant according to inclusion-exclusion criteria we defined. All papers have been classified according to research type contribution and research type facet. We mapped all papers according to their research contributions and we obtained a graph to assess the current research contributions and their trends in the future.

Our future work will be to complete this work by writing a survey paper to assess all possible solutions to secure and preserve-privacy in IOT environment.

REFERENCES

[1] Aaditya Jain, B. S. (2016, April). Internet of Things: Architecture, security goals, and challenges. International Journal Innovative Research in Science & Engineering (IJIRSE), Vol.No2:Issue4.

[2] Alfaqih, T. M., & Al-Muhtadi, J. (2016). Internet of Things Security based on Devices Architecture.International Journal of Computer Applications.

[3] Athreya, A. P., DeBruhl, B., & Tague, P. (2013). Designing for self-configuration and self-adaptation in the “internet of things" in Collaborative Computing: Networking Applications and Worksharing. 9th International Conference Collaboratecom, (pp. 585-592).

[4] Bagozzi, R. Y. (1991). Assessing Construct Validity in Organizational Research . Administrative Science Quarterly (36:3), pp 421-458.

[5] Bouij-Pasquier Imane, A. A. (2015). A Security Framework for Internet of Things. 14 th International conference, CANS 2015, , (pp. 19-31 Volume 9476 of the series Lecture Notes in Computer Science). Marrakesh.

[6] Burnett L., K. B.-S. (Volume 10, Issue 4, May 2003). The GeneTrustee: a universal identification system that ensures privacy and confidentiality for human genetic databases. Journal of law and medicine, 506-513.

[7] Cavalcante E. et al. (2016). On the interplay of Internet of Things and Cloud Computing: A systematic mapping study. Computer Communications Volumes 89-90, Pages 17-33.

[8] Charu C. Aggarwal; Philip S. Yu, eds. (2008). "A General Survey of Privacy". Privacy-Preserving Data Mining – Models and Algorithms

[9] Ding Chao, L. Y. (2011). Security Architecture and Key Technologies for IoT/CPS . ZTE Communication, 17(1):11-16.

[10] Erez Shmueli, T. Z. (2014). Constrained obfuscation of relational databases. Information Sciences, Volume 286, 35.

[11] Gang G., L. Z. (2011). "Internet of things security analysis," in Internet Technology and Applications (iTAP), 2011 International Conference on, 1-4.

[12] Gregor, S. (2006). The Nature of Theory in Information Systems. MIS Quarterly (30:3), 611-642.

[13] Hernandez-Ramos JosAl' L., J. B. (2015). Preserving Smart Objects Privacy through Anonymous. Sensors - Open Access Journal.

[14] Hevner, A. M. (2004). Design Science in Information Systems Research. MIS Quarterly (28:1), 75- 105.

[15] JianQiang Li, J.-J. Y. (2013). A top-down approach for approximate data anonymisation . Enterprise Information Systems, 272.

[16] Junqing Le, X. L. (2016). Full Autonomy: A Novel Individualized Anonymity Model for Privacy Preserving. Computers & Security.

[17] Kocher, P. L. (2004). Security as a new dimension in embedded. In: Proceedings of the 41st Annual Design Automation Conference, DAC 2004, San Diego, CA, USA, June 7-11 (pp. 753-760). New York: ACM.

[18] Liu C., Y. Z. (2012). Research on Dynamical Security Risk Assessment for the Internet of Things inspired by immunology, in Eighth International Conference on Natural Computation (ICNC).

[19] Leusse P, P. P. (2009). Security Cell, a security model for the Internet of Things and Services.International Conference on in Advances in Future Internet, (pp. 47-52).

[20] Loukil F., Ghedira C., Aïcha-Nabila B., Boukadi K., Maamar Z. Privacy-Aware in the IoT Applications: A Systematic Literature Review. International Conference on Cooperative Information Systems (CoopIS) 2017. Proceedings, Part I. Lecture Notes in Computer Science 10573, Springer 2017, ISBN 978-3-319-69461-0, Oct 2017, Rhodes, Greece.

[21] Mingqiang Xue, P. P. (2011). Distributed privacy preserving data collection. In Proceedings of the 16th international conference on Database systems for advanced applications.

[22] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, 2007, pp. 106-115.

[23] Pan Yang, X. G. (2013). A Privacy-Preserving Data Obfuscation Scheme Used in Data Statistics and Data Mining. IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, (p. 881).

[24] Pierangela Samarati and L. Sweeney. k-anonymity: a model for protecting privacy. Proceedings of the IEEE Symposium on Research in Security and Privacy (S&P). May 1998, Oakland, CA.

[25] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. In Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering (EASE'08), Giuseppe Visaggio, Maria Teresa Baldassarre, Steve Linkman, and Mark Turner (Eds.). BCS Learning & Development Ltd., Swindon, UK, 68-77.

[26] Philipp Offermann, O. L. (2009). Outline of a design science research process. In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST '09).

[27] Ricardo Neisse, G. S. (2015). A Model-based Security Toolkit for the Internet of Things.ScienceDirect.

[28] Robert Bredereck, A. N. (2014). The effect of homogeneity on the computational complexity of combinatorial data anonymization. Data Mining and Knowledge Discovery, Volume 28, Number 1, 65.

[29] Samani A., H. H. (2015). Privacy in Internet of Things: A Model and Protection Framework. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015) (pp. Volume 52, 2015, Pages 606-613). Procedia Computer Science.

[30] Shmatikov, J. B. (2006). Efficient anonymity-preserving data collection. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06). ACM, New York, NY, USA, (pp. 76-85).

[31] Syazarin, N., Aziz, N. A., Daud, S. M., & Syarif, S. A. (2017). An Overview on Security Features or Internet of Things (IoT) in Perception Layer. Journal of Engineering and Applied Sciences.

[32] Usha P., R. S. (2014). Sensitive attribute based non-homogeneous anonymization for privacy preserving data mining. International Conference on Information Communication and Embedded Systems (ICICES2014), 1.

[33] Venable, J. (2006). The Role of Theory and Theorising in Design Science Research . First International Conference on Design Science Research in Information Systems and Technology, (pp. 1-18). Claremont, CA: Claremont Graduate University.

[34] Xiao L, H. B. (2010). A knowledgeable security model for distributed health information systems.Computers & Security., (pp. 331-349).

[35] Xin Ma, Q. H. (2010). Study on the Applications of Internet of Things in the Field of Public Safety.China Safety Science Journal, 20(007):170-176.

[36] Yunjung Lee, Y. P. (2015). "Security Threats Analysis and Considerations for Internet of Things". 2015 8th International Conference on Security Technology (SecTech), (pp. vol. 00, no. , pp. 28- 30).

[37] ZhangW., B. Q. (2013). Security Architecture of the Internet of Things Oriented to Perceptual Layer.in International Journal on Computer, Consumer and Control (IJ3C), Volume 2, No.2.

[38] Zhiqiang Yang, S. Z. (2005). Anonymity-preserving data collection. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05). ACM, New York, NY, USA, (pp. 334-343).

AUTHORS

Moussa WITTI is a consulting engineer and IT architect in the R&D. He is advising bank and insurance firms in content and data management. He has more than 13 years of IT application development and deployment experience. He has obtained an MBA from Toulouse Business School and master Research in Computer Science from university of Franche-Comté in Besançon (FRANCE).

Dimitri KONSTANTAS is Professor at the University of Geneva (CH) and director of the . He has been active since 1987 in research in the areas of Object Oriented systems, agent technologies, and mobile health systems, with numerous publications in international conferences and journals. His current interests are Mobile Services and Applications with special focus in the well-being services for elderly and information security. Professor D. Konstantas has a long participation in European research and industrial projects and is consultant and expert to several European companies and governments.

INTENTIONAL BLANK

AN ASPECT-ORIENTED WITH BIP COMPONENTS FOR BETTER CROSSCUTTING CONCERNS MODERNIZATION IN IOT APPLICATIONS

Anas M. R. AlSobeh and Aws A. Magableh

Department of Computer Information System, Yarmouk University, Irbid, Jordan

ABSTRACT

The complexity of Internet of things applications is inherited from the nature of Internet of things components interactions, construction, and implementation of non-functional requirements (crosscutting concerns). Managing such complexity is extremely difficult since implementing crosscutting concerns tend to be spread out and tangled across core IoT architecture. In this paper, we propose an aggregated model of aspect orientation paradigm and BIP components to provide better means to deal with these complexities. Our proposed model provides IoT high level abstractions which gathers relevant contextual properties pertaining to the environment of IoT interactions. We integrate BIP components to generate solutions for a complex tracking and tracing logic of interaction characteristics that might provide better separation of concerns and modularization.

KEYWORDS

Aspect-Oriented Programming, Aspect Orientation, Modularization, Behavior-Interaction-Priority Model, BIP Components, Internet of things, IoT, crosscutting concerns, Aspects

1. INTRODUCTION

Internet of Things (IoT) is a modern technology evolved throughout the years. IoT has turned into the most crucial and prevalent system that empowers everybody to make, create, offer, utilize the data to produce information. IoT is the network of physical things, cars, water systems, home appliances and other items embedded with electronics, software, sensors with different types, and networks which enables these objects to connect and exchange data [3]. IoT targets at connecting of smart devices facilitating interactions among things and people. With IoT real word and digital words are interacting through various kinds of technologies such as internet protocols, sensors types and communication [4]. Having all these issues in mind, there will be so many crosscutting concerns that spread across different components of IoT architecture [15]. Managing and dealing with these crosscutting concerns in IoT environments is not an easy task, due to lack of management tools that ensure the performance, robustness, dependability, and security of IoT systems [2].

To address such issues, we need to introduce a dynamic management approach which provide better separation of concerns. Aspect Orientation (AO) is a suitable technique that introduce a modularization concept to encapsulate common crosscutting concerns into lossy coupled

Dhinaharan Nagamalai et al. (Eds) : CSEN, NCWC - 2018
pp. 21–31, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81203

abstraction [12]. However, Aspect-Oriented Programming (AOP) is limited to a set of specific programming constructs such as constructors, methods, and proprieties. Consequently, less attention has been paid on developing AO to be used for model based design flow. Behavior-Interaction-Priority (BIP) model has been designed to provide a formal description language for essential real-time interactions (e.g. IoT systems) between system's components based on a set of priority rules for the transition of the behavior [7].

In this paper, we present and discuss the modelling of a combination model-based design of both BIP and AOP (named BIP-AOP) to overcome the challenges of modularizing IoT-related crosscutting concerns. The proposed model presents BIP-AOP as an IoT-aware interaction design through introducing a set of intercepted invoke/execution points. These points represent IoT-related context information, which are encapsulated into a high-level abstract aspect. Our abstract aspect can be extended to create an efficient runtime crosscutting module. Such a module is implemented into runtime advices that might be woven into an IoT-based system dynamically. BIP-AOP consists of three-layer architecture: IoT layer, AOP layer and Application layer. IoT layer represents systems interactions between components; AOP layer represents a mediator on the IoT infrastructure to glue the IoT-related components with needed application concerns; and Application layer allows developer to customize the IoT behaviors without prior implementation knowledge. All those layers are mapped into BIP components to address the design of an event-based interaction components as a solution for pulling all relevant context information from IoT components.

The paper is organized as follows: section 2 provides an overview of the AOP and BIP components and a set of important related works, section 3 illustrates the motivated case study that inspired us to propose our model. Section 4 discusses our proposed model, finally conclusion and future work have been presented in section 5.

2. BACKGROUND AND LITERATURE REVIEW

Crosscutting concerns in network systems using IoT face challenges stemmed in nature of the interactions between nodes and resources. This makes the dynamic control and management at run-time is a fundamentally complex problem. Managing interpretability and complexity are one of the key essential ways of controlling the run-time interactions between connected nodes at different IoT layers architecture [1]. This section explains a little bit of background and essential major related studies and works that focused on managing IoT application complexity and portability either using AO or non-AO methods.

2.1 NUTSHELL BACKGROUND

The interactions amongst the different abstraction layers of the IoT-based application architecture impact the overall system's complexity and portability. IoT-based applications are overlapping networks of heterogeneous objects. Thus, the Representational State Transfer (REST) design is an architectural style that enables application-layer interoperability and reuse. Additionally, AOP is used to decompose systems, but the nodes may be tautly coupled in a design IoT-centric component, which is more often complex. Here, we use the most IoT systems support for the design REST-based applications. Understanding the below concepts are vital to understanding the novelty of the idea.

2.1.1 ASPECT-ORIENTED PROGRAMMING (AOP)

Generally, the relationships among user requirements and program components intersect or crosscut in distributed and extended systems. In other words, a requirement may have to be

implemented through several components; while on the other hand, a single component could cover more than one requirement or different parts of more than one requirement. Thus, a component may provide core functionality and at the same time include code for several other system requirements. An approach that tries to overcome this programming difficulty is called aspect-oriented software development (AOSD). These aspects encapsulate the functionalities that crosscut other functionalities in different parts of a system (Ex IoT systems). In AOSD, an executable AO program is created by automatically combining or 'weaving' together objects, methods and aspects to create a program that is not only easier to maintain, but also to reuse [5]. Aspect-oriented programming (AOP) is one of the most promising methods that developers can use to produce encapsulated objects that do not have any unnecessary additional functionality. This type of programming enables the developer to divide crosscutting concerns (i.e., an activity is also known as the separation of concerns (SoCs) into single logic, i.e., aspects. These aspects are the modular units of crosscutting concerns. In addition, as mentioned above, new behaviour can be added to a cloud application without the need to alter or interfere with the base source code. There are three key components in AOP: joinpoints, pointcuts, and advices [8] [9].

2.1.2 INTERNET OF THINGS (IOT)

Internet of Things has been defined as a paradigm consisting of a variety of uniquely identifiable day to day things communicating with one another to form a large scale dynamic network. The exponential growth in semiconductor domain has resulted in an explosion of usage patterns of cost-effective sensor based processor system. These systems when get empowered with advanced communication technologies (e.g., Bluetooth Low Energy, LoRA, ZigBee, Insteon, 3G, 4G, 5G etc.) converges into an emerging form of technological domain-Internet of Things or in short IoT. IoT aims to offer, a massive scale, heterogeneous, interoperable, and context-aware, and simplified application development cum deployment capabilities to the enterprises and end-users. The Internet of Things (IoT) envisions a world in which everyday objects collaborate using the Internet in order to provide integrated services for users. This vision defines the IoT as a dynamic global network requiring global self-managing capabilities, based on standard and interoperable communication protocols.

2.1.3 BIP COMPONENT FRAMEWORK

BIP stands for (Behaviour-Interaction-Priority) [6] is a formal framework for building complex systems by coordinating the behaviour of a set of atomic components. Behaviour is defined as a transition system extended with data and functions. The definition of coordination between components is layered: in the first layer lie the component interactions, while the second layer involves dynamic priorities between interactions [10].

2.2 RELATED LITERATURE REVIEW

Our literature review is focusing on investigating and reviewing the existing works those talks about utilizing AO in IoT systems, AO and BIP model in IoT and other approached were used to maintain portability of IoT applications. As stated in [4], there is the decent amount of works have been done to maintain service discover and service quality using different approaches, in [4] they have proposed AO to extend and enable IoT application to be more portable. They have stated that IoT derives various challenges from the Internet in the context of scalability, heterogeneity, undefined topology and data point information, incomplete metadata, and conflicts in user preferences. They have investigated the ability to develop an AO intermediate layer to inject the needed context related functionality. Their proposition works as a layered interface between IoT applications hardware and data gathered at software.

Few more works such as [1] have proposed a model-based design flow for networked systems with nodes running an Internet of Things (IoT) operating system. The design flow specifically targets web service applications of REST style and it is based on a formal modelling language, the BIP component model. Figure 1 explains their approach to managing IoT applications by using the BIP model. However, in [7] defined a method to modularize crosscutting concerns in the BIP component-based design. The authors have defied the BIP using AOP in a formal method and have given a mathematical representation for the AO and BIP concepts. However, this work has not proposed a solution for possibly modularization the IoT system concepts and related common crosscutting concerns.

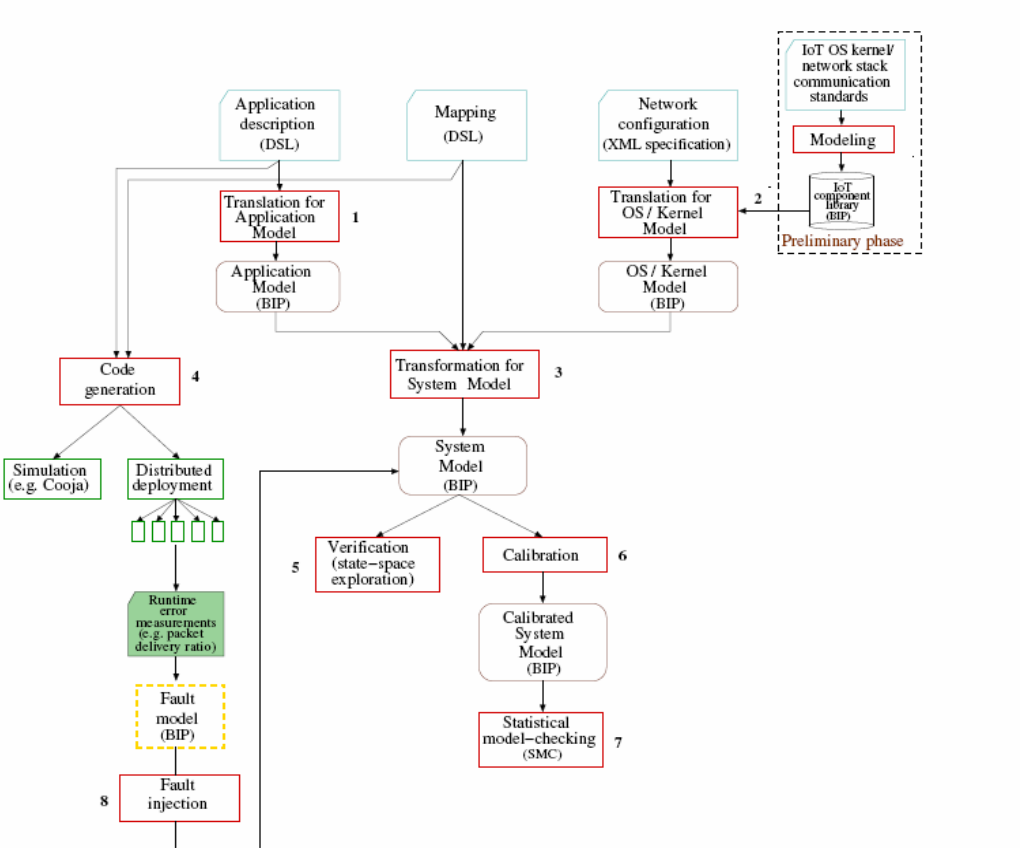

Figure 1. Model-based design flow for IoT [1]

Aspect-oriented programming is being used to manage systems (such as component-based systems) where IoT is considered as one of them. Some of the decent amount of works have been proposed at this space [11] [12].

3. CASE STUDY

IoT environments have a significant potential to provide for monitoring of water services to promote the possibility of tracing water flow. Such environments are typically equipped with many heterogeneous sensors that monitor both water and environmental parameters. The best example of it is water level display in the tank. IoT water system is used by water-level sensors to determine the level or amount of water that flow in an open or closed water system. Sensors usually detect the specific battery energy levels, if the sensing state is low then the brightness of dashboard light is reduced. It is integrated into the single device to get an alarm or trigger. These

sensors measure water levels within a specified range and continuously make a notification of the water the level.

Figure 2 illustrates basic IoT water system components that represent the dashboard that consist of data input and sound components. The energy management component is responsible for keeping track of power level for turning off or sleeping the dashboard and responsible for controlling on the level of dashboard brightness (LED) by observing the percentage of power volume.

Energy management is a common crosscutting concern and often poorly modularized in traditional design approach. Battery-level applications such as remote controls, dashboard, sound and sensor network can be preserved: (a-Low energy state) by reducing sound volume by 33% turn off dashboard after 10 seconds of no interaction; (b-medium energy state) by reducing sound volume by 25%, and reducing display brightness to 50% after 5 seconds of no interaction; and (c-high energy state) by reducing sound volume and brightness 0%. Implementation energy management component as an aspect-oriented extension that supports decomposition and integration of it with core IoT water system including that management as crosscutting. From secondary requirement perspective, focus on the energy management concerns that will manifest as interaction in the IoT system, can be encapsulated in loosely coupled aspect module.

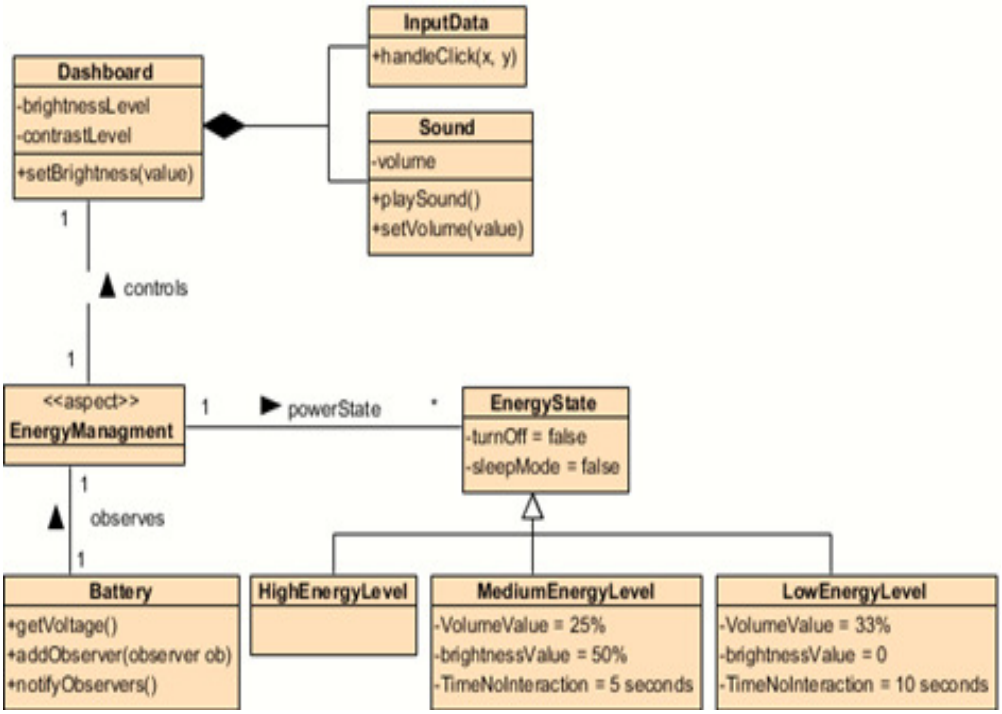

Figure 2. Basic IoT Components for Smart Water

Traditionally, *EnergyManagment* implementation is scattered across multiple classes (Dashboard and Sound) while calculation of energy state is tangled across multiple components. Indeed, the main difficulty is not the code complexity only, but it is mainly associated with interactions IoT nodes and component state they effect, which might refer to these interactions as “spaghetti bowl” [13] [14], as shown in Figure 3.

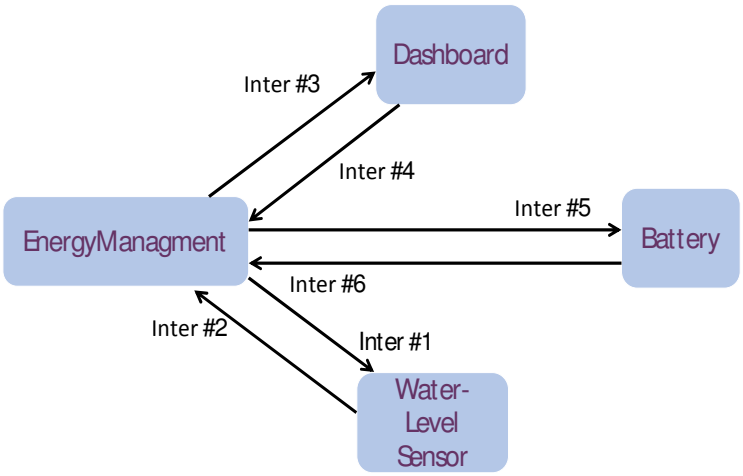

Figure 3. IoT Interaction Model

BIP is a model-driven engineering approach can help in reducing the complexity of IoT systems designs through its support a clear separation between architecture and interaction to allow for compositional design and analysis of systems [7]. However, BIP does not address the design of crosscutting concerns that do not manifest as code in the system but is complementary to existing techniques for capturing such requirements. In general, the code of such management interaction cuts across the system components, so their modularization significantly reduces their complexity, where the state of practice is to use BIP constraints for some behavior characteristics, BIP-AOP is an extension on top of such techniques to demonstrate its use to modularization and reuse. See next sections for more details.

4. PROPOSED BIP-IOT SYSTEM DESIGN USING AOP

During a comprehensive analysis of an IoT application, it emerged that concerns related to IoT architecture and behaviour were most significant. Inspiring BIP with the AOP technique was conceived to design and support the modularization during the development of IoT applications [7]. Our design has a formal semantics and makes a clear separation between IoT components to allow for decomposition IoT design into maintainable and reuse modules. Figure 4 shows context-aware aspects that is utilized to capture the behaviour and interaction concerns of IoT components. It presents components, interactions, priorities, and their composition. An atomic-context aspect is the basic low-level computation modules, which encapsulate IoT-related context information. It is implemented as an aspect and their behaviours defined as a glue-aspect that is extended with high-level abstractions. Transitions are represented as arrows associated with context-related IoT components to transfer data and messages between components.

Figure 4 demonstrates artifacts of the proposed aspect layer- model design, they are possible to generate the BIP-AOP components to be reused in the IoT application. The IoT components are instantiated from the IoT design definition including the IoT's application mapping onto AOP system components. Atomic-context components obtained by pulling a set of interactions among low-level components occur when execution of one component modifies the behaviour of another one. Such interactions may cause by modification or extension aspects to any state that is accessible by high-level glue-aspect components, including the state of the core IoT application, communications, protocols, and network.

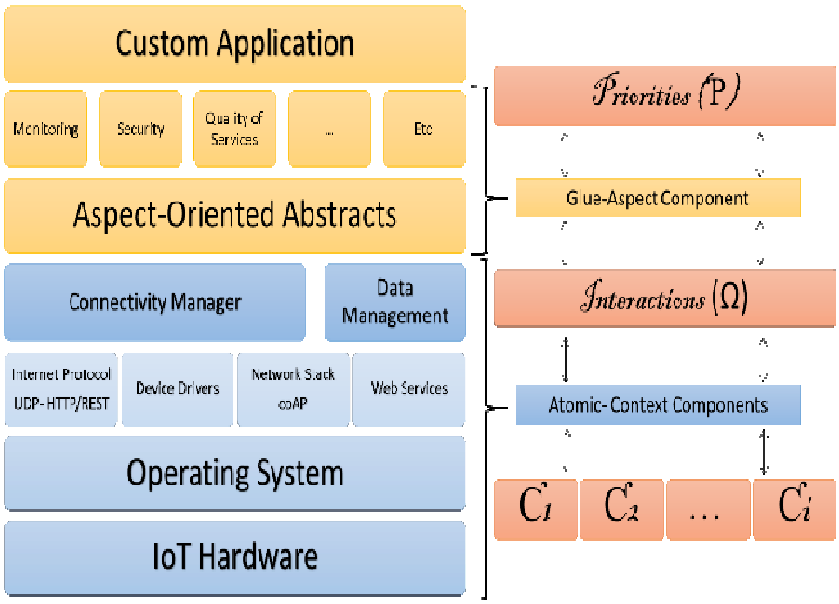

Figure 4. Standard IoT Design-based Model with AOP

4.1 THE BIP MODEL OF THE IOT APPLICATION

The overall our approach involves converting the application design definition for the IoT application and its protocol, e.g., REST, into BIP components as shown in Figure 5. Description of structure allows constructs application analysis findings back to the IoT design definition. The BIP components implemented by the BIP-AOP architecture are instantiated from the IoT framework through determining the components' context parameterization and their characteristics. These are encapsulated in the atomic-context component which is formalized as an observer for tracing the state space of secondary requirements, i.e., crosscutting concerns, with the BIP components as mentioned. Validation of properties derived from functional and non-functional IoT requirements takes place with by state-space exploration with BIP components according to the specified mapping onto system's component.

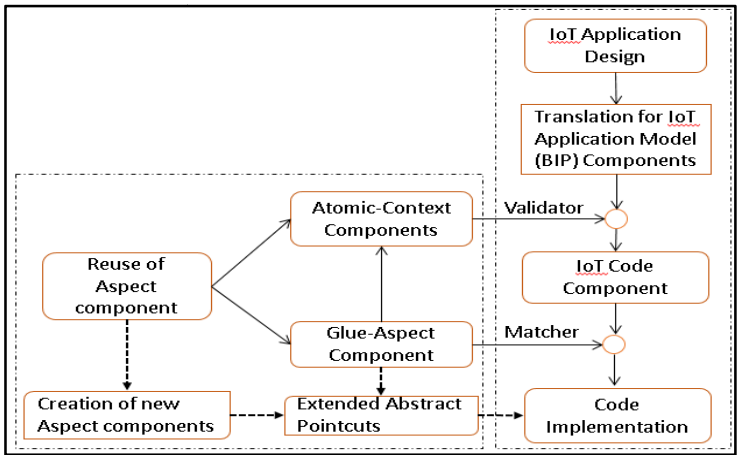

Figure 5. Architectural BIP-AOP for IoT Activities

The runtime properties IoT requirements take place by invoking the generated code on the components and within IoT environment. All IoT constraints are pulled and weaved to/into the core code modules. In AOP, for instance, abstract aspect used to impose constraints on the matching of pointcuts and application of advice. Glue-Aspect components can be framed as the abstractions directly representing the architecture of IoT event-based interactions as detailed below.

4.2 INITIAL CONTEXT-AWARE ASPECTS

Context primitives allow the IoT application developer to identify scattered and tangled over multiple codes using existing standards, technologies, and protocols for encapsulating IoT crosscutting concerns and leverage wherever possible [16]. BIP encourages the development of a smaller set of defacto standards for IoT concerns (e.g., implement policy and practice to ensure the monitoring concerns of IoT interactions of distributed components), an interaction is a software implementation based on behaviour function(s) that transforms groups of data into context data, and priority is a selection process, we select more sequence of execution flow of advices out of many other advices.

In context-aware IoT applications, Definitions 1&2 identify the semantic properties for support of the different type of information that allows applications to adapt their behaviour in response to interaction in the IoT environment using AOP. These properties are encapsulated in the aspect-oriented abstract layer.

Definition 1 ($Advice$): An advice A encapsulates a set of context data C_a is defined by a pointcut P and a set of joinpoint$the\ J_p\ such\ that \forall p \in \boldsymbol{P}: J_p \subseteq\ \boldsymbol{C}_a$.
Definition 2 ($Atomic\ Context\ Aspect\ Ap$): An abstract aspect is a tuple$\langle C_a | P | J_p \rangle$, where Ap is a transition state consists of relevant context proprieties, which is possibly receiving the new valuation of the J_p holds the application of computation $Advice$ with the set possible properties in $C_a\ of.$

Develop scalable approaches for separation of resource-constrained IoT nodes. We adapt some fundamental IoT-related context concepts into BIP components. IoT nodes interactions require that applications deal with the inherent unreliability of communications and processes. We have identified six primary context concerns that require support by developers. They are distributed, node discovery, limited connectivity, location, proximity, and quality of service. This context information can be exploited to address the semantic heterogeneities of data exchanged by nodes interactions, e.g., Atomic-context data of sensors, s1 → d1 means that sensor 1 has produced a piece of data that is numbered 1. Likewise, s2 → d2 means that sensor 2 has produced a piece of data that is numbered 2. Sensors will likely be heterogeneous, from different manufacturers, and collect data, with varying levels of data integrity. Usually, sensors are geographically located. Sensors may have an owner(s) who will have a control over the collected data, who can access it, and when. Implementing such input leads to scattered across multiple modules which cause spaghetti bowl as discussed in section 3.

Interaction Ω serves as the glue-aspect component that encapsulates crosscutting concerns through their atomic-context components. An interaction involves one or more abstract aspects of different atomic-context properties and extends advice that realizes data management between the IoT application components.

Definition 3 (*Interaction*): An interaction Ω is enabled iff its C_a holds and all its pointcuts are invoked. An invoked interaction is called from the complete list of possible interactions that based on the states of the atomic-context components.

The BIP approach defines the invoked interactions and executes its $Advice$, which defined in the application context with an atomic-context aspect that is extended by implementing the logic that interferes with the execution of an IoT-based component used a special kind of inner class. Hook atomic-context components invoke their corresponding pointcuts given the new value received by the designated joinpoints. In the following, we consider an atomic-context component $C_a$with behavior is filtered the invoked higher-level abstractions dynamically and decrease non-determinism.

Definition 4 (*Priority*): Priorityρ is a set of advices to be applied once a pointcut p has been matched to be found simply by specifying the precedence ordering to the abstract aspects, which contain the $advice$ in the glue-aspect components.

Apart from these changes, the weaving semantics for regular aspects does not have to be modified for IoT-based aspects. Priority ρ over C_a is used to define the event that should be performed preceding or succeeding a function execution. BIP-AOP model components are elaborated on AOP processes. It extends by defining glue-aspects for capturing multiple concerns and specifying a precedence order on the set of interactions Ω, which is defined the set of transitions satisfying Definition 4.

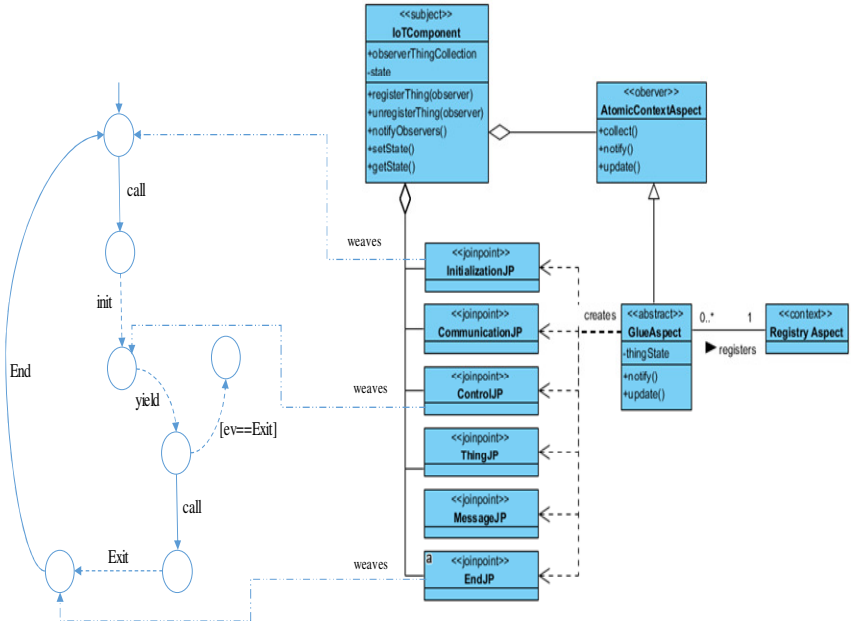

Figure 6. BIP-AOP Architectural IoT Pattern Design

4.3 BIP-AOP MODEL-BASED ARCHITECTURE

Figure 6 illustrates the architectural BIP-AOP design model in IoT domain. It traces state of every distributed IoT components based on using observing state approach. In case, the state of interaction changes, IoT components will be notified at different levels in the system. We define many BIP components to provide the understanding of using different IoT concepts (thing, protocol, node, communication, etc.) in an IoT context as discussed earlier. *AtomicContextAaspect* is an aspect component acts as an observer into a spectrum of the lower-level model to be a controller of the IoT components interactions. In other words, it represents the aspect that implements the advice function whose task is to handle all the crosscutting concern logic (e.g., monitoring, security, management, etc.). It relies on AOP inter-type declarations to

introduce context properties and methods to IoT components and APIs. We extend this aspect with the proposed high-level customization aspect called *GlueAspect*, to weave custom logic or amend the normal workflow of interaction in the system. The idea behind defining such aspect is to decouple the implementation by offering pointcuts, advices, inter-type declarations were defined in *AtomicContextAspect* to exclude components dependency. *GlueAspect* involves a set of joinpoints such as initialization, call and execution for the *InitializationJP, MessageJP, CommunicationJP, EndJP, ControlJP, ThingJP* and so on. *GlueAspect* offers pointcuts to pick up those joinpoints and then use advice to inject the logic of the crosscutting concerns. Once running the IoT application, the state of the one thing object has been changed by the BIP event handler (such as call, initiation, exit, etc.) all other components objects depending on it will be notified and take appropriate action as shown in Figure 6. The *RegistryAspect* then takes responsibility to register and monitor all IoT components interactions states.

5. CONCLUSIONS

This paper has proposed an integrated model of using AOP and BIP to provide a better modularization of cross-cutting concerns in of IoT applications. The proposed model demonstrates crosscutting issues in IoT application and highlights how AOP addresses them. Specifically, the purpose of this paper is to provide a roadmap for using BIP model components with AOP to deal effectively with crosscuts in IoT application design, systems interaction, and integration. In the future, we will work on conducting a primary experiment to showcasing the efficiency and effectiveness of using our model in encapsulating, modularizing and separating crosscutting concerns obliviously.

REFERENCES

[1] Lekidis, A., Stachtiari, E., Katsaros, P., Bozga, M., & Georgiadis, C. K. (2018). Model-based design of IoT systems with the BIP component framework. Software: Practice and Experience, 48(6), 1167-1194.

[2] Tselentis, G., & Galis, A. (Eds.). (2010). Towards the future Internet: emerging trends from European research. IOS press.

[3] Borgia, E (2014), "The Internet of Things vision: Key features, applications and open issues", Computer Communications,Vol.54, pp.1-31.

[4] Balakrishnan, S. M., & Sangaiah, A. K. (2015). Aspect oriented middleware for Internet of things: a state-of-the art survey of service discovery approaches. Int. J. Intell. Eng. Syst, 8(4), 16-28.

[5] Shanmughaneethi, V., Praveen, R. Y., & Swamynathan, S. (2012). CIVD: detection of command injection vulnerabilities in web services through aspect–oriented programming. International Journal of Computer Applications in Technology, 44(4), 312-320.

[6] Basu, A., Bensalem, B., Bozga, M., Combaz, J., Jaber, M., Nguyen, T. H., & Sifakis, J. (2011). Rigorous component-based system design using the BIP framework. IEEE software, 28(3), 41-48.

[7] El-Hokayem, A., Falcone, Y., & Jaber, M. (2016, July). Modularizing crosscutting concerns in component-based systems. In International Conference on Software Engineering and Formal Methods (pp. 367-385). Springer, Cham.

[8] Djoko, S. D., Douence, R., & Fradet, P. (2012). Aspects preserving properties. Science of Computer Programming, 77(3), 393-422.

[9] Katz, S. (2006). Aspect categories and classes of temporal properties. In Transactions on aspect-oriented software development I (pp. 106-134). Springer, Berlin, Heidelberg.

[10] Verimag: BIP Tools, http://www-verimag.imag.fr/BIP-Tools,93.html

[11] Nazarpour, H., Falcone, Y., Jaber, M., Bensalem, S., & Bozga, M. (2017). Monitoring Distributed Component-Based Systems. arXiv preprint arXiv:1705.05242.

[12] Ma, K., Sun, R., & Abraham, A. (2013). Toward a Module-centralized and Aspect-oriented Monitoring Framework in Clouds. J. UCS, 19(15), 2241-2265.

[13] Abbes, M., Khomh, F., Gueheneuc, Y. G., & Antoniol, G. (2011, March). An empirical study of the impact of two antipatterns, blob and spaghetti code, on program comprehension. In Software maintenance and reengineering (CSMR), 2011 15th European conference on (pp. 181-190). IEEE.

[14] Mikkonen, T., & Taivalsaari, A. (2007). Web Applications: Spaghetti code for the 21st century.

[15] Bilal, M. (2017). A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers. arXiv preprint arXiv:1708.04560.

[16] Perera, C., Zaslavsky, A., Christen, P., Compton, M., & Georgakopoulos, D. (2013, June). Context-aware sensor search, selection and ranking model for internet of things middleware. In Mobile Data Management (MDM), 2013 IEEE 14th International Conference on (Vol. 1, pp. 314-322). IEEE.

[17] AlSobeh, A., & Clyde, S. Unified Conceptual Model for Joinpoints in Distributed Transactions. In ICSE (Vol. 14, pp. 8-15).

AUTHORS

Dr. Anas AlSobeh, Dr. AlSobeh received his B.Sc. and M.Sc. degrees in computer information systems from Yarmouk University in 2007 and 2010, respectively. He also received PhD degree in computer science from Utah State University/USA with honour in Dec. 2015. He joined Yarmouk University academic staff in 2016. He is currently an assistant professor of computer information systems (CIS). His research interests include web technology, e-learning systems, software engineering, distributed systems, cloud systems, Internet of things (IoT) and data modelling. He has many scientific publications. He also has European-funded projects. He is also an active member of credit mobility projects to exchange academic members.

Aws A. Magableh is an Assistant Professor at the Faculty of Information Technology and Computer Science in Yarmouk University, Jordan. He obtained his PhD from the National University of Malaysia (UKM) in 2015, he obtained his master in Software Engineering from University Malaysia (UM) in 2008, and Bachelor's in Software Engineering from the Hashemite University in 2006. His research interests include Web technology, Web Services, Cloud Computing, Aspect-Orientation, Software Engineering, System design and modelling. Aws is very passionate about Learning & Development (L&D) and I have been immersed in the training industries with Nokia, Microsoft and Huawei for the past 10 years focusing on building skillsets

INTENTIONAL BLANK

ASPECT-ORIENTED SOFTWARE SECURITY DEVELOPMENT LIFE CYCLE (AOSSDLC)

Aws A. Magableh and Anas M. R. AlSobeh

Department of Computer Information Systems, Faculty of Computer Science and Information Technology, Yarmouk University, Irbid, Jordan

ABSTRACT

Recently, the need to improve the security of software has become a key issue for developers. The security function needs to be incorporated into the software development process at the requirement, analysis, design, and implementation stages as doing so may help to smooth integration and to protect systems from attack. Security affects all aspects of a software program, which makes the incorporation of security features a crosscutting concern. Therefore this paper looks at the feasibility and potential advantages of employing an aspect orientation approach in the software development lifecycle to ensure efficient integration of security. It also proposes a model called the Aspect-Oriented Software Security Development Life Cycle (AOSSDLC), which covers arrange of security activities and deliverables for each development stage. It is concluded that aspect orientation is one of the best options available for installing security features not least because of the benefit that no changes need to be made to the existing software structure.

KEYWORDS

Aspect Orientation, AO, Aspect-Oriented Programming, AOP, SSDL, Software Security Development Life Cycle, Security.

1. INTRODUCTION

The software development life cycle (SDLC) of an information system (IS) consists of four main stages: planning, creating, testing, and deployment. It has also been described as involving a requirement, design, coding, and documentation phase. The SDLC is applicable to a variety of configurations because an IS can comprise just hardware, just software, or both [3]. Given the current global situation and the heightened need for security in both industry and government as well as in personal life, are search area that is growing in importance is the enhancement of the SDLC to include the implementation of the security software development life cycle (SSDLC).

Some of the recent security threats and attack reports can be found in [19] and [20]. A more comprehensive analysis of the exploits, vulnerabilities, and malware based on data from Internet service providers and over 600 million computers worldwide can be found in [1]. Figure 1 illustrates the attacks that focused on applications during the period of 2016. According to [1], "Disclosures of vulnerabilities in applications other than web browsers and operating system applications decreased slightly in first half of 2016, but remained the most common type of vulnerability during the period, accounting for 45.8 per cent of all disclosures for the period."

Dhinaharan Nagamalai et al. (Eds) : CSEN, NCWC - 2018
pp. 33–47, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81204

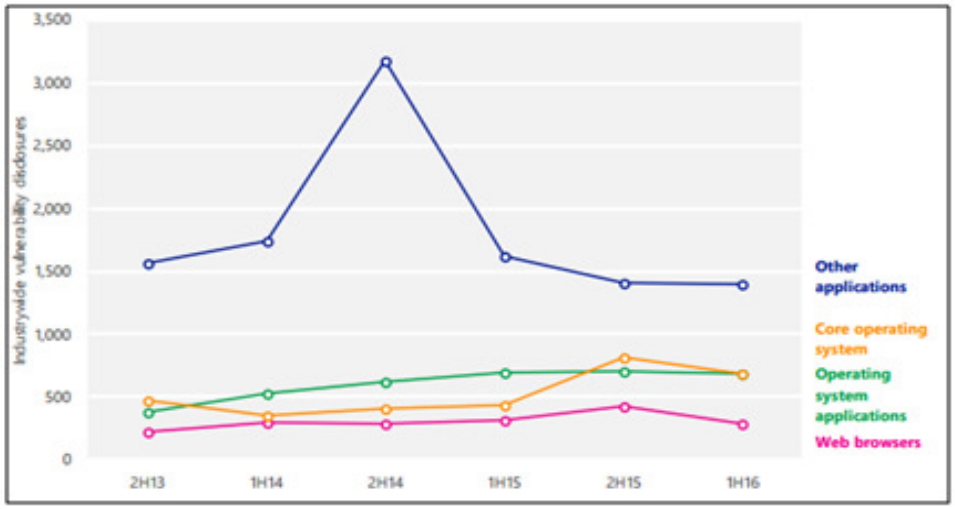

Figure 1: Example of a Microsoft Security Intelligence Report [1]

Thus, it can be said that security is the main requirement of all users, especially those in charge of critical infrastructure. Therefore it is crucial that software vendors address the issue of security threats head on. However, creating software that is ever more secure is a huge challenge [2]. Nevertheless, software vendors must endeavour to do so in order to maintain society's trust in computers in this digital era One of the key steps that software vendors and their collaborators need to take is to shift to a substantially more secure SDLC process that places a greater emphasison security in order to reduce the amount of vulnerabilities in all stages of the process– from requirement to documentation–and that attempts to reduce such vulnerabilities as early in the SDLC as practicable.

The SSDLC helps developers build more secure software and address security compliance requirements. It is created by adding security-related activities to any stage of the software development process by incorporating the concept of aspect orientation (AO) into the SDLC, as shown in Figure 2.

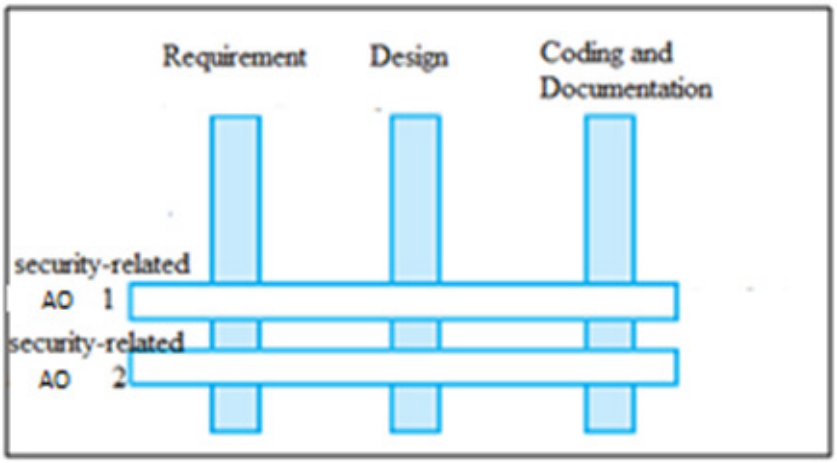

Figure 2: Inclusion of security activities in the SDLC

The two main goals of this study are to identify the techniques currently being used to enhance the security of the SDLC through an in-depth review of the literature and to propose a model to enhance the security of software development. The main objective of this study is to utilize the strength of AO and its concepts to enhance software development security. This work aims also to eject security activity into SDLC with less amount of impact on the standard process of development. The study was guided by two research questions: "What is the practical applicability of existing models for a secure software development life cycle?"and "How can aspect orientation enhance SSDLC?"

The remainder of this paper organized as follows: section 2 provides an overview of the key concepts addressed in this paper. Section 3 explains the methodology used in this research. Section 4reviewsrelated works. Section 5 explains the proposed aspect-oriented software security development life cycle (AOSSDLC) model, and finally, section 6 contains a conclusion and suggestions for future work.

2. OVERVIEW

Before discussing the research methodology, an overview of the key concepts of AO and SSDLC is provided in order to clarify their contribution to the main aim of this study.

2.1 ASPECT ORIENTATION

A research team headed by Gregor Kiczales at Palo Alto Research Center coined the term 'aspect-oriented' when they were developing aspect-oriented programming (AOP) as well as AOP language (AspectJ),a language that is now very popular among developers working in Java [21]. Just as object-oriented (OO) programming [22] before it resulted in a wide range of OO development methodologies [23], AOP has engendered a growing number of software engineering technologies such as AO development methods, modelling techniques that are usually based on the principles of unified modelling language (UML) [24], and assessment technologies to test the effectiveness of AO approaches. Nowadays, the term 'aspect-oriented software development (AOSD)' is used to refer to an array of software development techniques that support the modularization of aspects (also known as crosscutting concerns) throughout an entire software system [25]. This modularization covers requirement engineering, business process management, analysis and design, and programming. Aspect orientation offers a systematic means by which to modularize crosscutting concerns which, as the name implies, has an effect on the other concerns. More often than not, crosscutting concerns cannot be smoothly or completely decomposed from the rest of the system in the design or in the implementation phase, which means that, at implementation, scattered code (code duplication) and/or tangled code (significant dependencies between concerns) can appear. Figure 3 illustrates both of these problems: Figure 3(a) shows how the logging aspect code can get scattered and duplicated in other concerns while Figure 3(b) shows how that same code can become tangled up in one concern.

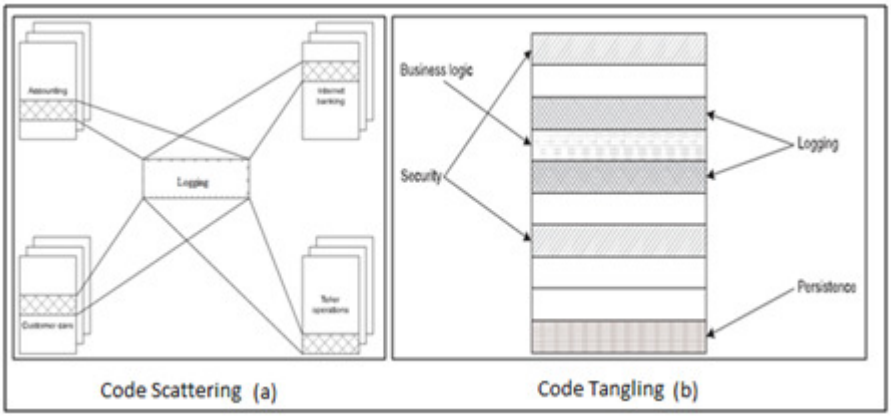

Figure 3: Code scattering and code tangling

2.2 SOFTWARE SECURITY DEVELOPMENT LIFE CYCLE

During the SSDLC the processes of software development are modified by embedding activities that result in enhanced software security. This section summarizes these activities and the ways in which they are incorporated into a SDLC in general terms. Here, the SDLC consists of requirement, analysis, design, and implementation. Also, it should be noted that the modifications are not designed to completely change the developmental process, but to add clear deliverables for software security. Moreover, the software architecture should be designed in such a way that the software is able protect not only itself but the data it processes [2]. Hence it is important that designers assume that security faults will exist in a system and that software should therefore run with the least privileges. Services that are not regularly needed should be disabled by default or made accessible to just a few users or distinct groups of users [2]. Also, tools and guidance should be provided with software at deployment to help users and administrators use it securely, and updates should be easy to deploy. The implementation of security measures in the SDLC is not limited to the above; it needs to be considered in the requirement, implementation, verification and release stages as well.

3. METHODOLOGY

As mentioned in the introduction, this study attempts to answer the following questions:

RQ1. What is the practical applicability of existing models for a secure software development life cycle?

RQ2. How can aspect orientation enhance the SSDLC?

RQ3: What would be the impact of employing AO and its concepts to enhance the security of the SDLC?

In order to find answers to the above questions, we analysed a large number of research papers that were published during the period January 2003to November 2017. The year 2003was chosen as the start date because it was during that year that publications on the security of the software development life cycle began to appear. The papers were collected by using three complementary

search methods in order to achieve the maximum coverage of the domain, as illustrated in Figure4. First, we performed a manual search of the proceedings of several conferences which are particularly relevant to the SDLC. Second, we searched through a number of digital libraries. Third, we performed a snowballing search [34] on the papers collated by the first two search methods. Snowballing involves retrieving papers that are cited by the considered papers (forward snowballing) and papers that cite the considered paper as a reference (backward snowballing). For forward snowballing, we used the bibliographies of the identified papers. Google Scholar was used to perform backward snowballing; Figure 4 illustrates the search process.

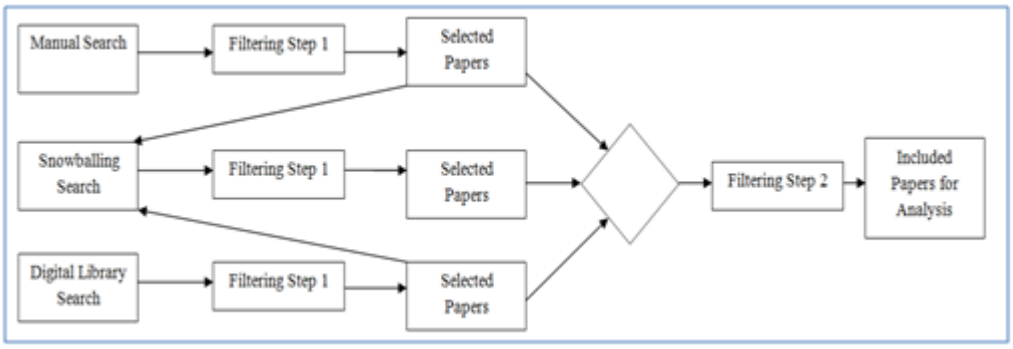

Figure 4: Search strategy

4. LITERATURE REVIEW

Before presenting the proposed model, this section reports the results of our review of literature that was undertaken to discover whether any of the existing models use AO in the SSDLC. Literature review consists of reviewing, extracting and evaluating and then analyzing and interpreting the studies that are relevant to this research. Most research starts with a literature review. However, unless a literature review is fair, it is of little scientific value [4]. With the extended article of this work, we will be adopting a systematic literature review approach [5] we were able to find the best and most-cited works that are relevant to the research question posed by this study, i.e. What would be the impact of employing AO and its concepts to enhance the security of the SDLC? It is worth noting that research on the use of AO to improve the security of the SDLC is quite scarce. This section is exploring major works and categories of works that have been done to employ security in SDL.

4.1 MICROSOFT SECURITY DEVELOPMENT LIFE CYCLE (MSSDLC)

One of the first initiatives in relation to the SSDLC was the MSSDLC proposed by Microsoft, which works in line with the phases of a classic SDLC. Microsoft proposed some of the best security practices to fit with each stage of its classical SDLC, which is shown in Figure 5 [6].

Figure 5: Classical software development life cycle of Microsoft [6]

For the training stage, which is the initial stage of the classic SDLC at the company, Microsoft proposed some core security training to secure the other upcoming stages and focused particularly on the issue of privacy. At the stage of Requirements, Microsoft proposed quality gates and privacy risk assessment to determine on the privacy impact rating. The same thing goes to the other stages, for the design phase, Microsoft proposed quite a few security practices such as threat modelling and attach surface analysis. As for implementation, they suggested using approved tools, deprecating unsafe functions and performing static analysis. For all stages, best practices were applied to ensure the highest possible levels of security. Figure 6provides a summary of the main security activities Microsoft deployed for each SDLC stage [10].

Figure 6: Secure software development process model of Microsoft [10]

4.2 MODEL-DRIVEN ARCHITECTURE FOR THE SECURE SOFTWARE DEVELOPMENT LIFE CYCLE

A model is a crucial component of the design process in many engineering disciplines, including software engineering, as it represents a real system or entity and enables developers to test a proposal or prototype before expending a significant sum on the real thing[10]. To facilitate the software engineering process, the Object Management Group (OMG) developed model-driven architecture (MDA) (The OMG describes itself as an international, open membership, non-profit computer industry consortium and it came into being in 1989.)It is essentially a methodology that helps to specify software system specifications regardless of the hardware and platforms being used for the implementation. In MDA, there are three default models (CIM, PIM, and PSM), as shown in Figure 7 [7]. The concept has been involved in model-driven security (MDS), which can model security requirements at a high level of abstraction.

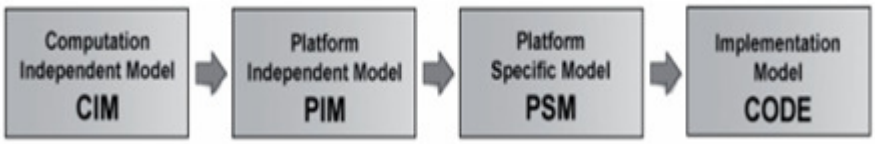

Figure 7: Structure of model-driven architecture [11]

Following the MDA for SDLC proposed in [9], an architecture known as MDA-SDLC was proposed [8], which considers security requirements, security models and system requirements and modelling throughout the SLDC stages. It illustrates the main roles and responsibilities along with technical skills set needed for requirements and software model requirement. At design model, architectural model and pattern model is suggested. The rest of stages contain few more activities suggested at each stage as illustrated in Figure 8.

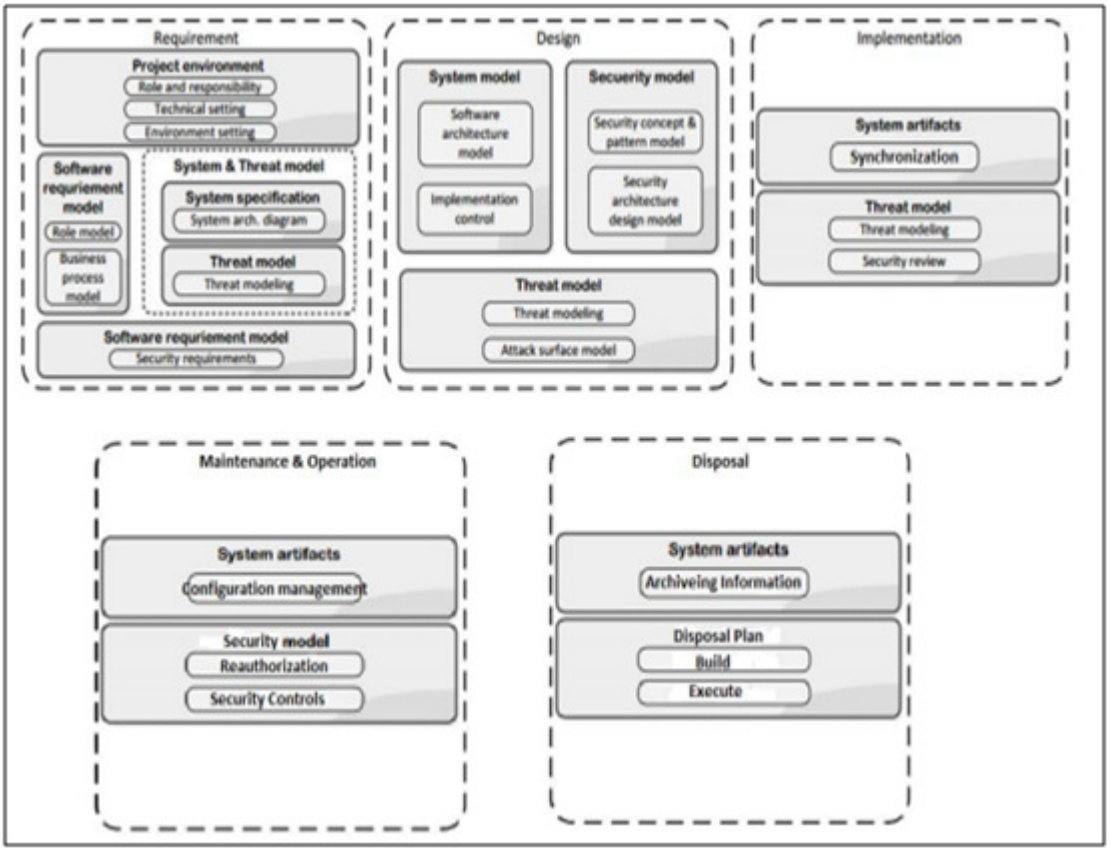

Figure 8: Overview of architecture of MDA SDLC [8]

A number of formal methodologies have been proposed to aid in the development of secure software systems and they cover several different SDL stages. The methodologies are mathematically based on specifications that represent software system behaviours. The specifications themselves employ a formal syntax and can be used to garner key information about a software system. Developers can produce software programs in a formal manner by using a formal methodology [7]. An illustration of a typical formal methodology is provided in Figure 9.

With respect to the SDLC, there are two types of formal methodology that can be employed: the software security assessment instrument method and the construction method. The first type involves the development and use of tools and a variety of information resources to ensure the security of software, for example by using model checkers. The second type uses a range of formal methodologies for the entire SLC, including formal description language for the system

specification and unequivocal programming language and bug prevention fixes, which enables analysis of the software to be very stringent in even the earliest stages of software development.

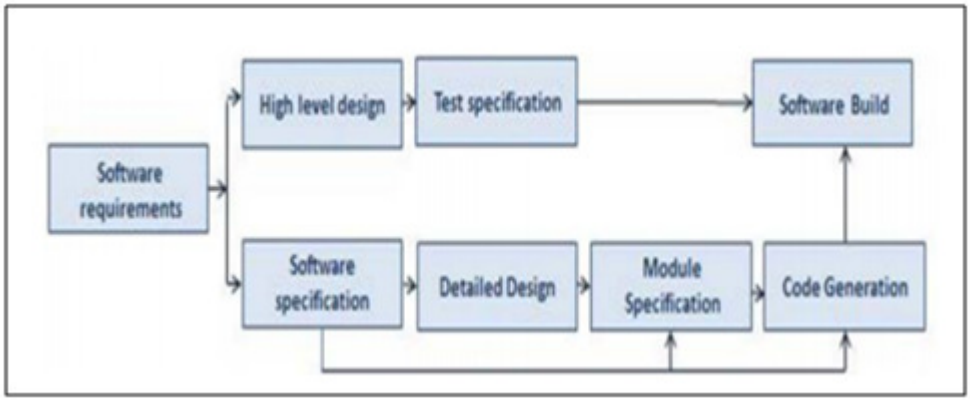

Figure 9: Application of typical formal methodology to SDL

4.4 ASPECT-ORIENTED MODELLING FOR REPRESENTING AND INTEGRATING SECURITY CONCERNS IN UML

This study suggests a new way of specifying security aspects in UML and, moreover, it enables security aspects to be systematically and automatically woven into UML. The main aim is to identify and deal with security-related concerns and model them during the design stage of the SDLC. To do this, it uses the class diagram as one of the UML design diagrams. Figure 10 shows the security aspect by using the class diagram.

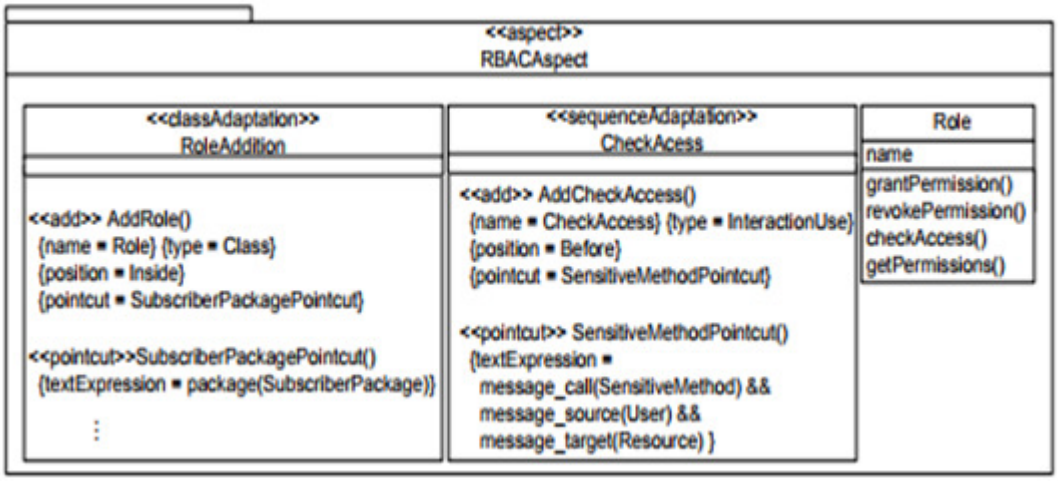

Figure 10: Class diagram of security aspect

Other approaches, such as [14], implement a risk management analysis in order to incorporate security into the SDLC. Other related works such as [27-32] have attempted to improve security by using AOP at the SDLC implementation stage. Moreover, among these works [27] and [28] have also proposed a method to integrate security using AOP at the implementation stage, while [29] and [30] have investigated aspectizing security at the programming stage. Additionally, [31]

and [32] have considered using AOP only during the programming stage to ensure that the system is trustworthy during the development process. Generally, less attention has been given to utilizing the benefits of AO and its related concepts for other (earlier) SDLC stages as a means to improve the security of software. However, it makes sense to employ AO as early as possible in the SDLC, otherwise it might be too late to address all the security dimensions.

5. ASPECT-ORIENTED SOFTWARE SECURITY DEVELOPMENT LIFE CYCLE

This section focuses on describing the architecture of the AOSSDLC proposed in this paper. As mentioned in the introduction, finding ways to ensure the highest level of security during the development of complex software systems is now more critical than ever because software now pervades almost all aspects of our lives both professional and personal. Aspect orientation has been shown to be effective in dealing with the crosscutting nature of security requirements so it could be particularly useful not only when developing and designing applications but also when implementing them. In our work, we aim to bring different fields together to discover whether the AO concept can be successfully utilized to improve SDLC security and consequently reduce security-related attacks and vulnerabilities. Figure 11shows how we see the various fields linking together.

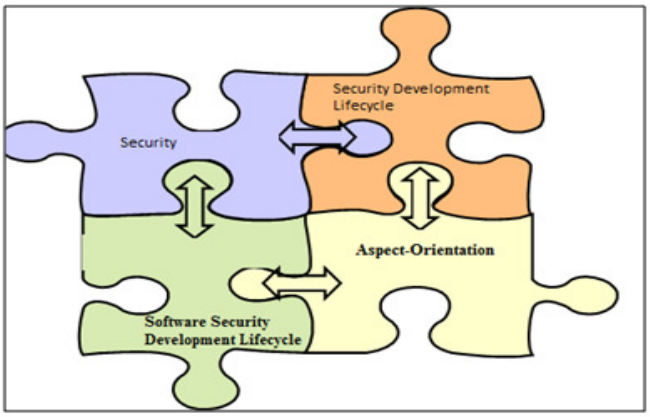

Figure 11: AOSSDLC areas of collaboration

Perhaps the major benefit of using AO is that it can weave any kind of crosscutting concern, including security and security-related concerns, into a system, even if they are scattered and tangled [26] throughout the system, without having an adverse effect on other concerns. Moreover, this weaving process can occur at any stage in the SDLC. In other words, adding or removing aspects in any stage of the SDLC becomes less problematic and less time consuming [15][17]. One more dimension to consider which motivated this work is that, the evolving field of securing SDLC [1] [8]. Having said that, it was essential to investigate and explore the ability to connecting these topics with each other to come up with relativity reliable secure software development life cycle depending on AO. Our proposed model, the AOSSDLC combines these advantages and addresses these issues. Figure 12 illustrates the application areas of the proposed model.

In the design, development and implementation of a secure system the security-related properties in the system should be abstracted out of the main system to improve clarity, maintainability, manageability and reuse [18]. Also, where legacy source code has identified or potential security

vulnerabilities the code should be patched by adding the smallest possible amount of new code and ideally the original code should not be changed. In addition, where appropriate, it should be possible to reuse security-related properties in a range of applications [28]. It is noteworthy that all the above can be achieved by using AO [18] because AO automatically checks for errors in security-sensitive calls, automatically logs data on security concerns, replaces generic code with secure code and specifies privileges, abstracts some concerns, replaces concerns with the minimum changes necessary and its changes are reusable in any stage of the SDLC. Thus, it becomes clear why our model is designed in accordance with the concept of AO in order to attempt to integrate the above mentioned highly desirable security-related activities into the SDLC. In AO there are two kinds of crosscutting concerns (aspect concepts) that applicable to any stage of the SDLC: dynamic crosscutting and static crosscutting. They can both be utilized to embed security-related crosscutting concerns into any part of SDLC. These types of crosscutting are described in brief in the following subsections.

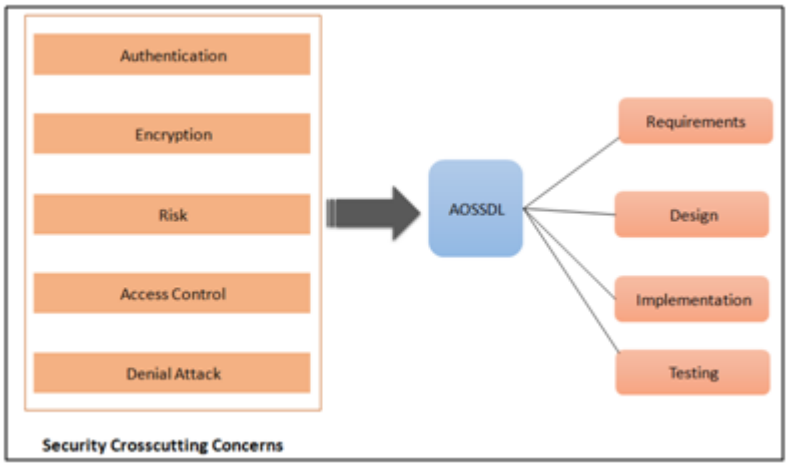

Figure 12: Application areas of proposed AOSSDLC model

5.1 DYNAMIC CROSSCUTTING OF AOSSDLC

Dynamic crosscutting is a technique that allows points to be defined and changes (pieces of code) to be recommended in the SDLC coding stage of the dynamic execution of a program. Our proposition extends this dynamic crosscutting technique to other SDL stages. Our proposed model utilized the dynamic crosscutting sub concepts join point, pointcut, and advice not only in the programming stage but in all the other SDLC stages as well in order to include dynamic security-related crosscutting concerns in the SDLC. In the AOSSDLC model, join points are predictable points in the execution of a program, and they are the points at which security activity must be added at a specific SDLC stage to ensure the security of the software. It is the pointcut in the AOSSDLC model that is designed to identify and select the join points where the security activity needs to be added. When the model gives advice this refers to the model identifying the actual security activity that needs to be injected and executed when a join point is reached. Figure 13 illustrates the proposed AOSSDLC model.

Join points are predictable point in the execution; it would represent the point where we would need to add the security activity at a specific SDL stage. Pointcut of AOSSDL is designed to

identify and select join points where the security activity will be added. Advice of AOSSDL is the actual security activity to be injected and executed when a join point is reached.

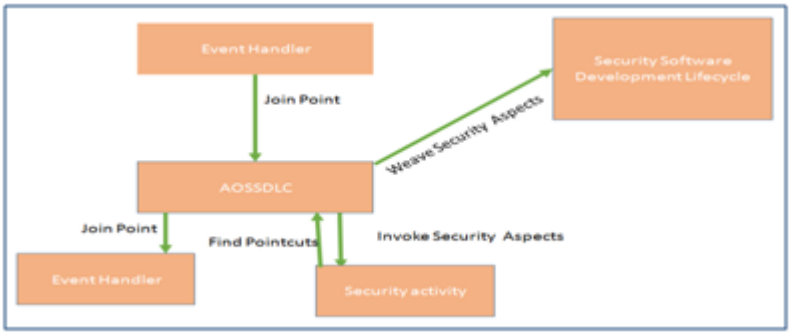

Figure 13: AOSSDLC model

5.2 STATIC CROSSCUTTING IN THE AOSSDLC

Static crosscutting is an inter-type declaration that can add attributes and/or methods to an existing structure. It is a powerful technique because it provides the developer with the capability to add new attributes and operations to a class or aspect, as well as a whole range of other declarations that affect the static-type hierarchy. In our proposed model, we use it to add attributes to a specific classification in the SDLC.

5.3 ASPECT WEAVING STEP

In this model, it is the weaving of the security aspects into the SDLC that makes the SDLC secure. This is done mainly by the following generic steps:

1) Locating the security join points, which involves identifying the locations at which the SDLC stage/activity and the security requirements/design aspects interact; analysing the vulnerabilities of and the threats posed to the software based on the security requirements and security design; and specifying the security join point setting for the connectors in the SDLC.

2) Constructing security advice, for which actions are defined in order to enforce security in the required SDLC stage through locating the join points that have the same vulnerability and grouping them together as a pointcut.

3) Weaving the security aspect into the SDLC in order to incorporate the security aspect into the SDLC, this involves systematically searching for join points so that the security advice/aspect can inject the required security behaviours into the SDLC in the correct places.

From a comparison of the proposed model with those in the literature, it would seem that the proposed model is better structured because not only does it have clear steps for defining changes in security at specific points in the SDLC stages, it also contains a weaving step to enable the injection of aspect changes. Moreover, the proposed model is based on a bottom-up technique

that maps the AOP elements (such as AspectJ) so that they can be embedded into the early stages of the SDLC.

5.4 APPLICATION OF THE AOSSDLC

The AOSSDLC model shows how the aspects are used and woven to inject a change and a modification at any stage of the SDLC without the need to work manually on the change. Figure 14 illustrates how AO is used in each stage.

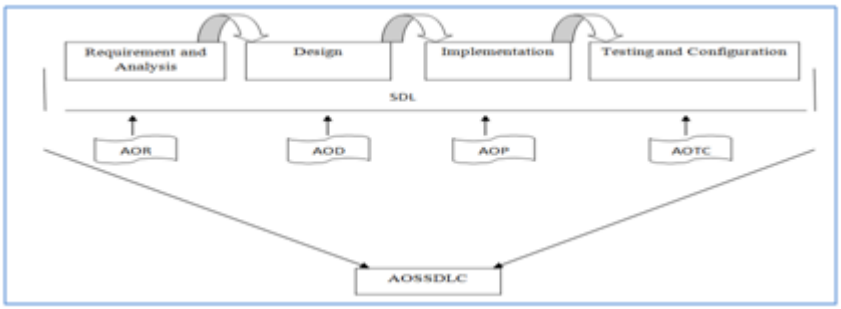

Figure 14: AO in the SDLC stages

Due to the limitation of space, here we illustrate how AO can be used in the requirement stage of the SDLC only. If the requirements have already been elicited from the clients, no major changes should need to be made to the actual requirements. Where a change does need to be made to a specific part of the natural text/requirement this will result in a change to another requirement because all requirements are connected and traceable. When there is a need to make such a change manually, the AOSSDLC model suggests utilizing the dynamic concept and structure of the aspects and aspect weaving, as shown in Figure 15.

```
Aspect RequirementChangeAspect {
        At Poincut RequirementChange (InitiateRQChangeJP _RQx): Execution
        (RequirementChangeAspect.Begin(..) && Args (RQx) )
        BeforeAdvise (InitiateRQChangeJP) {
                Check and add any prerequisite before changing and amending
                requirement/s
                Proceed (_InitiateRQChangeJP)}
        AroundAdvise (InitiateRQChangeJP) {
                Check and add roles of amending the requirement/s
                Proceed (_InitiateRQChangeJP)
                }
        AfterAdvise (InitiateRQChangeJP) {
                Add the rules of after changing and amending requirement/s
                Proceed (_InitiateRQChangeJP)
                }
}
```

Figure 15: Example of using of aspect orientation to make a change at the requirement stage of the software development life cycle

In light of the above discussion, we believe that this study was able to achieve its main objectives of identifying what models are currently being used to secure the SDLC and how AO can be used to make the software development process more secure.

6. CONCLUSION AND FUTURE WORK

This paper proposed an AO-based model for embedding security activities in the SDLC. Our ultimate aim is to develop a model that is practical and extensible for different SSDLCs and research projects. Our next step is to apply the AOSSDLC model to some real-life case studies, which will help us in assessing its performance in terms of how it deals with threats, the nature of its limitations, and its potential for scalability. The results, outcomes and feedback will be used to enhance the model and improve its feasibility and, consequently, promote its usage. We also intend to investigate the usage of AO in the so-called agile SDLC because this type of development life cycle has less stringent guidelines for the initial stages of development and then adjustments are made as and when needed throughout the remainder of the process, which is AO kind of behaviour where it does not affect any other processes and stages.

REFERENCES

[1] Microsoft. Microsoft security intelligence report: Julydecember 2016. http://www.microsoft.com/technet/ security/default.mspx, 2016

[2] Http://msdn.microsoft.com/en-us/library/ms995349.aspx#sdl2_topic1_1, retrieved July 2017

[3] Selecting a Development Approach, Retrieved May 2017.

[4] Ebse, B. & Charters, S. 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering.

[5] Maryati, Y. 2011. Systematic Review Hand-on Workshop. Research Center of Software Technology and management, Faculty of information and Technology, UKM.

[6] Howard, M., &Lipner, S. (2006). The security development life cycle (Vol. 8). Redmond: Microsoft Press.

[7] Sasikala D, The Most Common Methodologies for Secure Software Development, International Journal of Multidisciplinary Research and Development, Volume 3; Issue 3; March 2016; Page No. 194-197

[8] Muhammad Asad, Shafique Ahmed, Model Driven Architecture for Secure Software Development Life Cycle, International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 6, June 2016

[9] Zhendong Ma, Christian Wagner, Arndt Bonitz, and Thomas Bleier , Model-driven Secure Development Life cycle, International Journal of Security and Its Applications Vol. 6 No. 2, April, 2012

[10] Microsoft, Security Development Life cycle SDL Process Guidance Version 5.2, May23, 2012

[11] F. Truyen, "The Fast Guide to Model Driven Architecture; The Basics of Model Driven Architecture," Cephas Consulting Corp , 2006

[12] Eoin Keary, Jim Manico, "Secure Development Life cycle," in OWASP, Hamburg

[13] Richard Kissel, Kevin Stine , Matthew Scholl , Hart Rossman , Jim Fahlsing , Jessica Gulick, "Security Considerations in the System Development Life Cycle," National Institute of Standards and Technology , Gaithersburg, 2008 .

[14] Bart De Win , Riccardo Scandariato, Koen Buyens, Johan Gre´goire, WouterJoosen, On the secure software development process: CLASP, SDL and Touchpoints compared, Information and Software Technology 51 (2009) 1152–1171

[15] Georg, G., Ray, I., and France, R., "Using Aspects to Design a Secure System", In Proc. 8th IEEE Int'l Conf. on Eng. of Complex Computer Systems (ICECCS'02), pp. 117-128, 2002.

[16] Shah, V. and F. Hill, "An Aspect-Oriented Framework", In Proc. DARPA Info.Survivability Conf. and Exposition (DISCEX'03), pp. 22-24, 2003.

[17] Yu, H. et.al., "Secure Software Architectures Design by Aspect Orientation", In Proc. 10th Int'l Conf. on Eng. of Complex Computer Sys (ICECCS'05), pp. 45-57, 2005.

[18] Viega, J., Bloch, J.T., and P. Chandra, "Applying Aspect Oriented Programming to Security", In Cutter IT Journal, 14(2):31-31, 2001.

[19] Symatec, Internet Security Threat Report, Volume 21, April 2016

[20] Cisco, Cisco Annual Cybersecurity Report 2017 Infographic

[21] Schwanninger, C., &Joosen, W. (2011).Transactions on Aspect-Oriented Software Development VIII (Vol. 6580). S. Katz, & M. Mezini (Eds.). Springer Science & Business Media.

[22] Smith, B. (2015). Object-Oriented Programming. In Advanced ActionScript 3(pp. 1-23). Apress.

[23] Avison, D., & Fitzgerald, G. (2003). Information systems development: methodologies, techniques and tools. McGraw Hill.

[24] Magableh, A., Shukur, Z., & Ali, N. M. (2013). Aspectual UML approach to support AspectJ

[25] Filman, R., Elrad, T., Clarke, S., &Akşit, M. (2004). Aspect-oriented software development. Addison-Wesley Professional.

[26] Groher, I., &Voelter, M. (2007, March). XWeave: models and aspects in concert. In Proceedings of the 10th international workshop on Aspect-oriented modeling (pp. 35-40).ACM.

[27] B. De Win, B. Vanhaute, B. De Decker, "Security through Aspect-Oriented Programming", Advances in Network and Distributed Systems Security, pp. 125-138, 2001.

[28] J. Dehlinger and N. V. Subramanian.Architecting Secure Software Systems Using an Aspect-Oriented Approach: A Survey of Current Research. Technical report, Iowa State University, 2006

[29] De Win, Bart, WouterJoosen, and Frank Piessens."Developing secure applications through aspect-oriented programming." Aspect-Oriented Software Development (2005): 633-650.

[30] Marchand de Kerchove, F., Noyé, J., &Südholt, M. (2013, March). Aspectizing javascript security.In Proceedings of the 3rd workshop on Modularity in systems software (pp. 7-12).ACM.

[31] Safonov, V. O. (2008). Using aspect-oriented programming for trustworthy software development (Vol. 5).John Wiley & Sons.

[32] Mourad, A., Laverdière, M. A., &Debbabi, M. (2008). An aspect-oriented approach for the systematic security hardening of code. computers & security, 27(3), 101-11

[33] Mouheb, D., Talhi, C., Nouh, M., Lima, V., Debbabi, M., Wang, L., Pourzandi, M.: Aspect-Oriented Modeling for Representing and Integrating Security Concerns in UML. In: R.Y. Lee, O. Ormandjieva, A. Abran, C. Constantinides (eds.) Proceedings of the ACIS Conference onSoftware Engineering Research, Management, and Applications, Studies in Computational Intelligence, vol. 296,pp. 197–213. Springer (2010)

[34] A. van den Berghe, R. Scandariato, K. Yskout, W. Joosen, "Design notations for secure software: a systematic literature review", Software & Systems Modeling, 2015.

AUTHOR INDEX