https://doi.org/10.52131/jcsit.2020.0101.0004

*i*RASD Journal of Computer Science and Information Technology

Volume 1, Number 1, 2020, Pages 33 - 44

Journal Homepage:
https://journals.internationalrasd.org/index.php/jcsit

Evaluation of Model-driven Architecture for System-of-Systems Interoperability

Ahmad Ali[1], Nadeem Akhtar[2]

[1] Department of Information Technology (IT), The Islamia University of Bahawalpur, Pakistan.
Email: Nadeem.akhtar@iub.edu.pk
[2] Department of Information Technology (IT), The Islamia University of Bahawalpur, Pakistan.

ARTICLE INFO

Article History:
Received: August 20, 2020
Revised: October 26, 2020
Accepted: December 30, 2020
Available Online: December 31, 2020

Keywords:
System-of-Systems (SoS)
The Common Object Request Broker Architecture (CORBA)
Model Driven Architecture (MDA)
Common Warehouse Metamodel (CWM)
Meta-Object Facility (MOF)
PIM
CIM
Meta-Object Facility (MOF) is an Object Management Group (OMG) standard for model-driven engineering.

ABSTRACT

System-of-Systems consists of a number of task-oriented and dedicated systems working together combining their resources and capabilities in an efficient manner. The heterogeneity among systems of a System-of-Systems (SoS) makes interoperability an important issue. The interoperability issues between systems can create information, communication, or data loss between the collaborative systems. It is important to have a property-preserving information exchange between the collaborative systems of a System-of-System. A model driven approach is proposed in order to evaluate the interoperability issues of System-of-System. This paper analyzed the interoperability evaluation models for SoS based on Model Driven Architecture (MDA). The main objective of MDA is to design once and implement multiple times. The proposed approach uses CORBA, CWM, and MOF in order to collaborate between different types of operating systems and hardware platforms. The study based on smart-homes elaborates that not only our approach enhances the interoperability among SoS components using the formulation of business knowledge throughout a Model-Driven Architecture approach, but also opens the new horizons of formal basis for describing and scrutinizing the designs

Corresponding Author's Email: Nadeem.akhtar@iub.edu.pk

1. Introduction

Since the invention of computer systems another term is inextricably linked with it is, "Software" and each software must have a software architecture. Therefore, our focus is to discuss these terms one by one, as they are related to each other and when we moved towards SoS we must face a problem of collaboration among different system, which is called "Interoperability".

Engineering is a discipline that uses different techniques to get the quality results by utilizing the maximum available resources with cost and time effectiveness. To achieve this a systematic approach is followed to self-control every part of software development.

With the passage of time, humans make considerable developments in every field of life, as from the beginning of the time they moved towards modern life style, which is far developed from the past of humanization. In the beginning the humans lived in caves, have not any sense to wear cloths, have not civilized but now a days humans are much civilized. As the humans became more civilized, it also increases complexity in life like population, pollution, rush of traffic etc. Similarly, in the field of computer sciences humans also became civilized and to fulfil these modern needs of life they are indulged in the complexities of computer system, so that they have to move towards a brand new concept in computer science called "System of Systems".

The term "System of Systems" emerged in the past decade because humans moved rapidly towards the globalization in past few decades. This globalization of world compelled humans towards complex and more complex systems gradually and eventually led the basis of system of "System of Systems".

With the passage of time as we are rushing towards most real time complex systems, we are facing more complex situation among collaborating system due to lack of mutual relations among the working independent system. The one major issue we are facing while implementing system of systems is the "Interoperability" among the individual systems. The basic determination of the research is the evaluation of MDA (Model Driven Architecture) for SoSI (System of Systems Interoperability), which become more prominent for SoS. When we dive into the immense oceans of computer systems we've confused about the architectures of the systems that which system is considered as SoS, as "A collection of task-oriented or dedicated systems that pool their resources and capabilities together to obtain a new, more complex 'meta-system' which offers more functionality and performance than simply the sum of the constituent systems" (Popper, Bankes, Callaway, & DeLaurentis, 2004).

Maier defines a specific criterion for system of systems known as Maier's criteria (Maier, 1998).He pointed out five unique characteristics that must carried by system of systems. Any complex system failing to fulfil these characteristics will not fall in the category of system of systems. These are Operational independence; Managerial independence; Geographical distribution; Evolutionary development; and Emergent behavior.

In the light of this criteria:
"A system of systems consists of multiple, heterogeneous, operationally, distributed, occasionally independently, operating systems embedded in networks at multiple levels that evolve over time" (DeLaurentis, 2007).

2. Related work

It is human nature that seeks more and more, He put a noose on every aspect of the universe and exactly the same way many peoples of the past put their efforts on different aspects of this research. Let have a cursory look in history.

2.1. Software Architecture

In some previous decades, an important sub discipline of software engineering is arisen as software architecture. Architecture is the overall concept of a computer based system with its logical organization. In my viewpoint the architecture is fundamentally a Skelton that brings a base for the establishment a comprehensive design. It provides a structure to establish the base of design of software.

The use of different architectural descriptions while documenting the software describes the importance of software architecture. Some frequent descriptions of the system are "client-server organization," "layered system," "blackboard architecture," etc. (Clements, Garlan, Little, Nord, & Stafford, 2003)

The architecture is basically a skeleton that provides all the bases to create a complete design. Software architecture is just like human skeleton that defines the boundaries & shape of the human body and on the base of these boundaries, a complete human design is created. So, an overall view of software that did not provide actual view but complete shape, that's why Different designs are created from a single architecture just like different human beings have different shapes (designs) but have same skeleton (architecture).

2.2. System of Systems (SoS)

Our SoS surrounds four different aspects of software engineering, therefore we have to discuss about all the work related to these aspects. To extravagant the term SoS there is a need to set the point of view that discriminate the word system and System of System. Both expressions imitate to the recognized definition of system in that each consists of chunks, affiliations and a whole that is greater than the sum of the chunks, and therefore in that scenario they are the identical. (Boardman & Sauser, 2006)

Some work on the architectural qualms and establish that there is a necessity to create anSoSE management framework based on the demands of persistent technological progress in a complex dynamic environment. (Gorod, Sauser, & Boardman, 2008)

2.3. Model-Driven Architecture (MDA)

A methodology based on the RM-ODP that falls under the MDA initiative describe its principles by illustrating them with an example. (Gervais, 2002)

Some previous researches show different techniques to of model driven architecture that elaborated the importance of MDA for Distributed Applications. (Siegel, 2005)

Model Transformation is the core concept of MDA that introduces new approach that provides a new way of transformations from CIM-to-PIM and identified the core components features that can be used as the key kernels. (Zhang, Mei, Zhao, & Yang, 2005)

2.4. Interoperability

Some research done on the belief that interoperability need to happen at numerous ranks in and through agendas. The Software Engineering Institute with different perspectives defined a new model named "SOSI" come into existence.

"Achieving interoperability involves changes to the way the DoD does business, including: acquisition practices and guidance, technologies, engineering and management practices, operational doctrine for both the war fighter and those who support the systems. Joint Vision 2020 provides further challenges for the future. Realizing this vision requires that we begin to define approaches and models in more concrete terms". (Morris, Levine, Meyers, Place, & Plakosh, 2004)

In distributed systems the interoperability is a vital problematic issue and an ever tougher for increasing more levels of heterogeneity. Complexity is nowadays approaches to such a level that current methodologies are insufficient and that a most important a new dimension of thought is necessary to define doctrines to accomplish the vital problem of distributed systems. Findings suggest that evolving a third party as a middle man is an approach forward. (Blair et al., 2011)

2.4.1. Levels of Information System Interoperability (LISI)

A usually standard archetype for system of systems is LISI that concentrates on the increasing levels of complexity of SOSI.
Level 0: "Isolated interoperability in a manual environment between stand-alone systems"
Level 1: Connected interoperability in a peer-to-peer environment
Level 2: Functional interoperability in a distributed
Level 3: Domain based interoperability in an integrated environment
Level 4: Enterprise-based interoperability in a universal environment

2.4.2. Organizational Interoperability Maturity Model

It is the extension of LISI model that abstract all the layers of into five levels are:

Level 0: independent
Level 1: ad hoc

Level 2: collaborative
Level 3: integrated (also called combined)
Level 4: unified

2.4.3. The System of Systems Interoperability (SOSI) Model

SOSI model combines the technical interoperability with operational interoperability with the enhancement of programmatic anxieties among interoperable systems.

SOSI introduces three kinds of interoperability:

- **Programmatic**: How different programs interoperate to each other?
- **Constructive**: How different organizations interoperate to construct and maintain a system
- **Operational**: How different systems interoperate? (Morris et al., 2004)

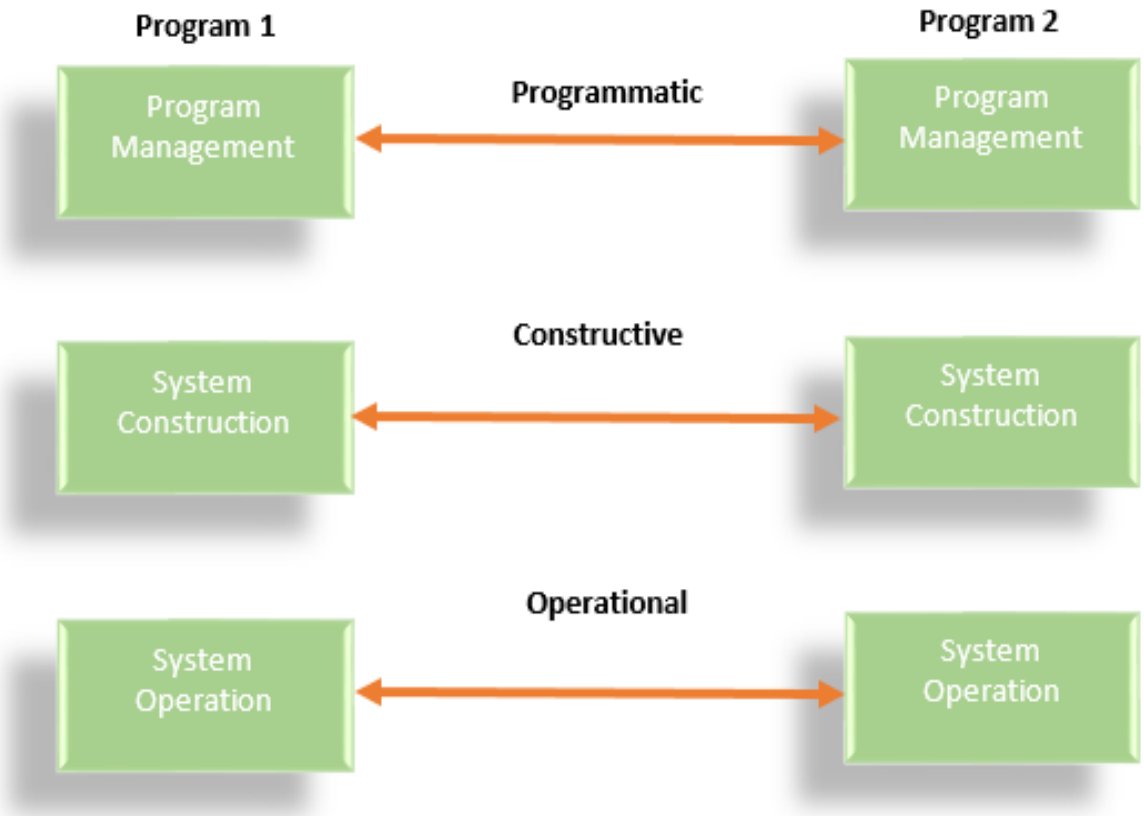

Figure 1. Different Types of Interoperability

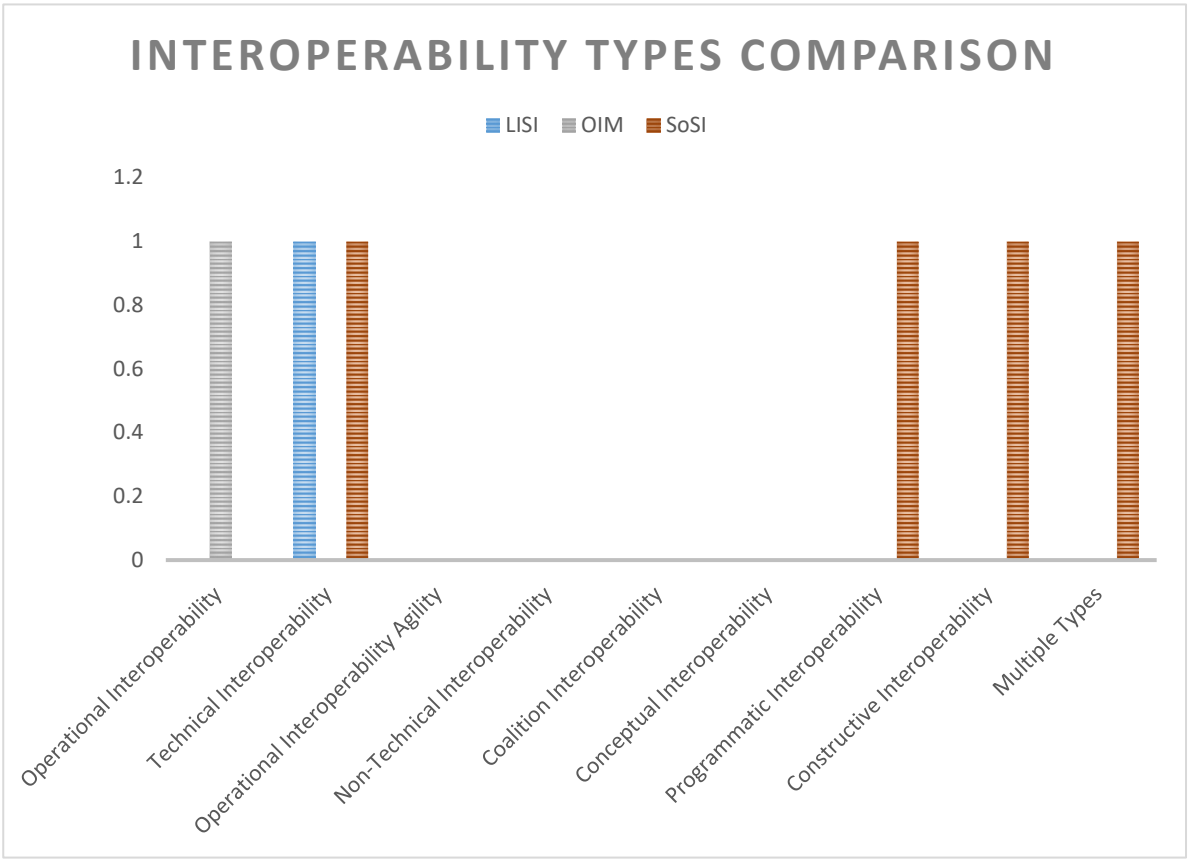

Figure 2. Protuberant Models Comparison

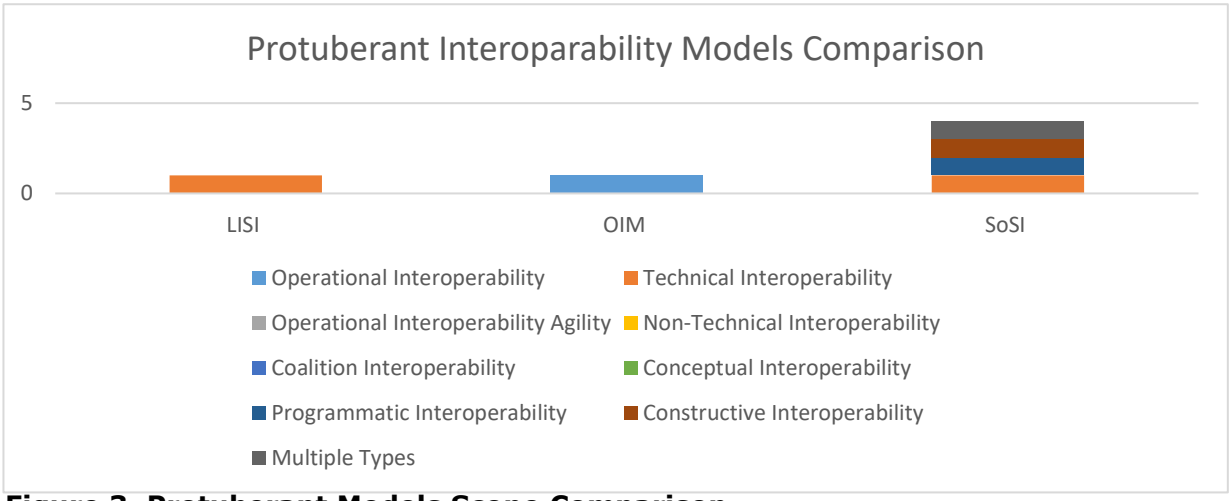

Figure 3. Protuberant Models Scope Comparison

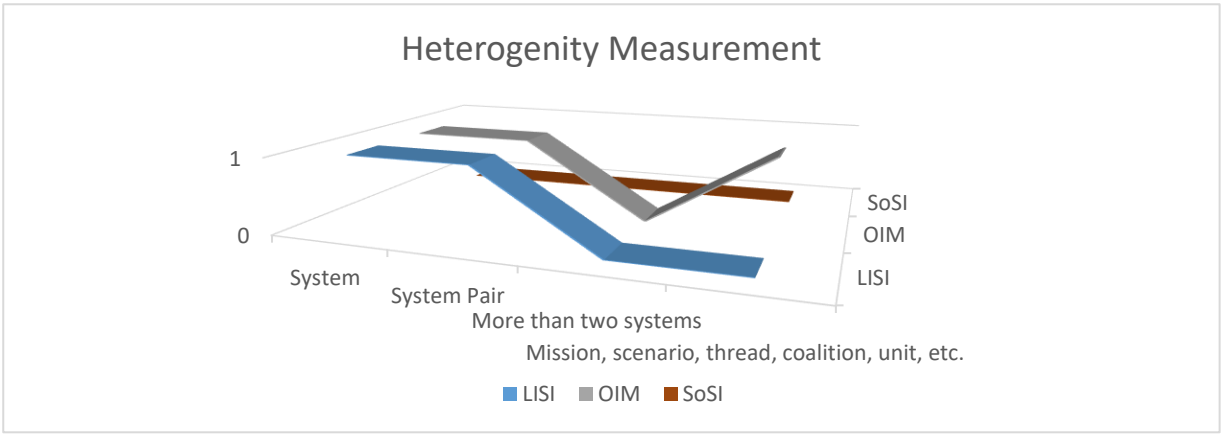

Figure 4. Type Measurement of Selected Models

The above scenario is in the favor of SOSI model but in terms of various system support the results are:

As the System of Systems (SoS) has to deal with diverse and heterogeneous systems and all the models are partially successful in different aspects of interoperability. SOSI model has some extension but still there is needed to be a model that successfully collaborate all of the proportions of interoperability.

3. Case Study

3.1. Home Automation

With the dawn of the emerging research, home automation become more prominent but it is a big challenge for developers to handle immense heterogeneity among systems & ultimately it become more sluggish for interoperability as interoperability is the technique to establish a communication between two (or more) systems to interchange info and data. Interoperability evaluation model development among software and information systems is challenging, and flattering a significant experiment.

The smart home heterogeneous environment encompasses:

- Security Management
- Room Management
- HVAC System
- Fire Safety System
- Energy Management
- Audio/Video Control
- Motion Detection
- Smoke Detection

Let's take it as case study for the research as smart home automation system comprises of these independent heterogeneous systems and it fulfills all the characteristics of SoS.

All the systems are totally different in terms of hardware, software and operating systems that ultimately leads towards the complexity of interoperability implementation. Heterogeneous systems with diverse stipulations are illustrating that the smart homes are moving towards data-intensive environment, as a result a few operative complications are introduced. The first problem is diverse heterogeneity and the second problem is the interoperability. (Perumal, Ramli, Leong, Samsudin, & Mansor, 2010)

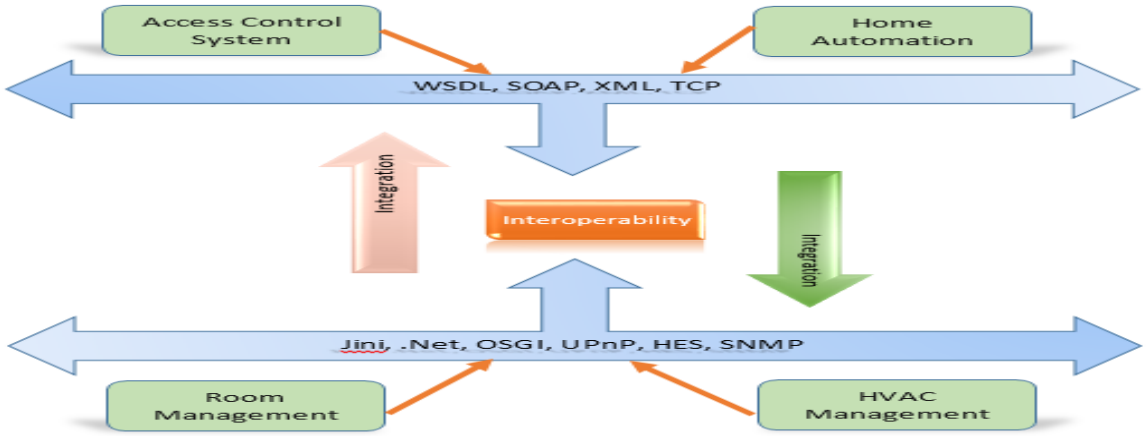

Figure 5. Interoperability among Heterogeneous Systems

This figure graphically describes the interoperability of smart home SoS in which the three different levels of interoperability tier from general OSI model are characterized.

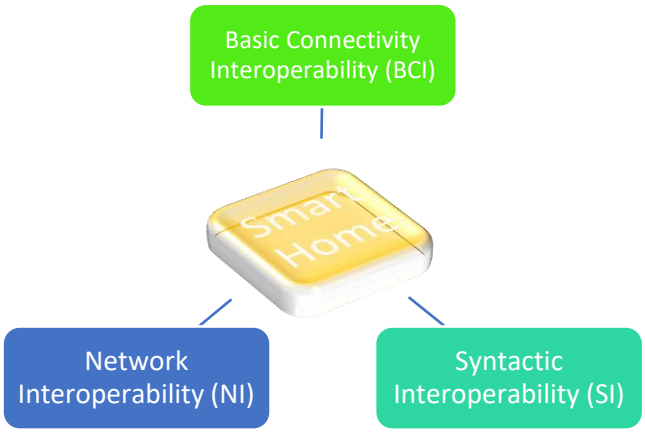

Figure 6: Interoperability Levels for Smart Home Environment

They derived interoperability levels as:

- Basic Connectivity Interoperability (BCI)
- Network Interoperability (NI)
- Syntactic Interoperability (SI)

3.2. Enterprise Architect

Considering the above-described interoperability levels a decision is taken to design a software model for smart home system of systems using the methodologies provided OMG as MDA. There are different organizations in the market that are claiming to provide a smooth environment to implement MDA's techniques for software designing. After critical observation a decision is taken to use Enterprise Architect by Sparx Systems as modeling tool to design smart home system of systems model.

3.3. Smart Home SoS CIM

Keeping in view these features of Enterprise Architect a Computational Independent Model (CIM) using Use Case Diagram Notations for smart home System of Systems is designed. The CIM is Basic MDA Model (View point) to describe software specifications at higher level of abstraction by hiding all technical & internal details. Let take a look at Computational Independent Model (CIM) for Smart Home System of Systems (SoS).

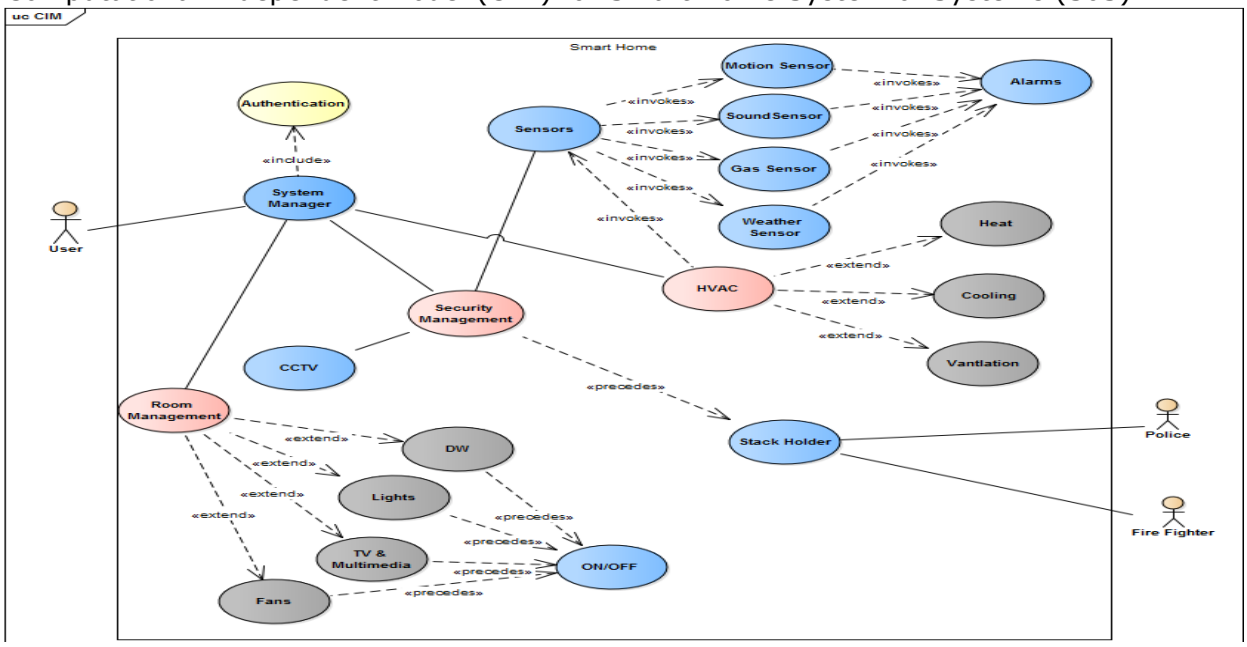

Figure 7. CIM for Smart Homes SoS

3.4. Smart Home SoS PIM

The main beauty of MDA is PIM showing all the architecture of Smart Home SoS with a high level of abstraction, define standards for all platforms to exchange data among heterogeneous resources.

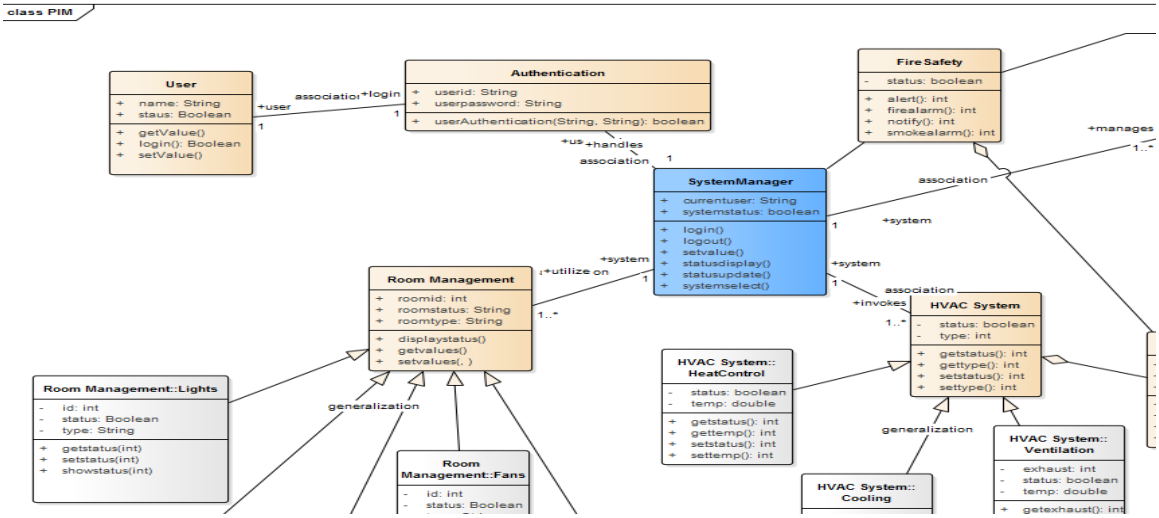

Figure 8. PIM for Smart Homes SoS

3.5. Smart Home SoS PSM

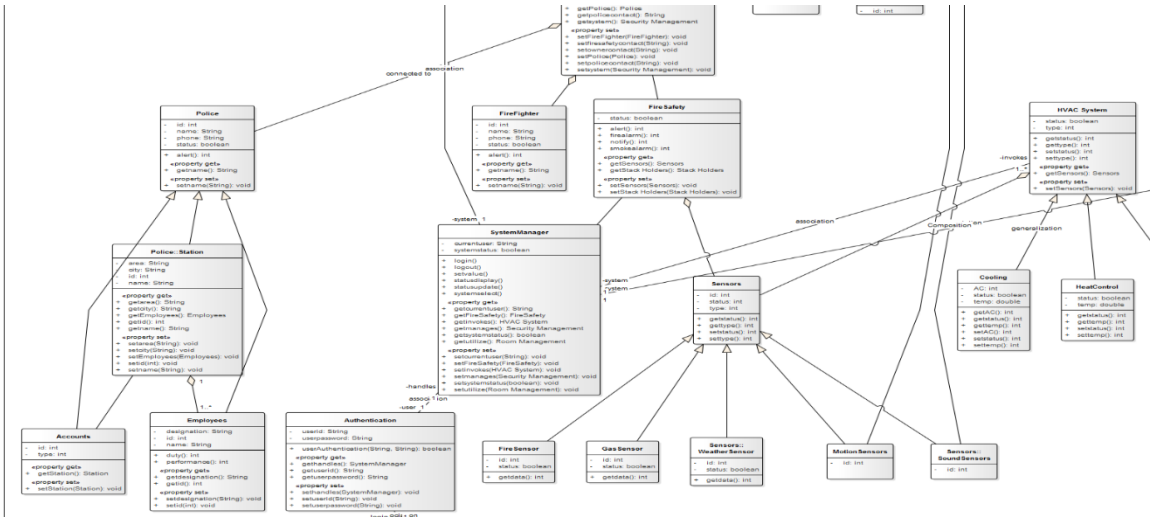

Figure 9. Portion of Java PSM for Smart Homes SoS

3.6. Simulations

MDA also provides Simulations for that we created a simple process activity diagram and run simulations. Take a look:

3.7. Code Generation

Not only providing the model transformation MDA also defines standards for code generation. We can directly generate programming language code from a PSM. Model Driven Architecture proving standards, Abstraction, Meta Models, Platform Independent Model to Platform Specific Models and Platform Specific Models to Platform Specific Models and all these conversions are made using different standards like Web Services Descriptive language (WSDL), Common Object Broker Architecture (CORBA), Extensible Mark-up Language (XML), Java Native Interfaces (JNI), Common Ware House Mode (CWM), Meta Object Facility (MOF) and many more.

4. Evaluation of Case Study

During the transformation of PIM to PSM and to Code Generation it seems to be that all three proposed interoperability types can easily be achieved due to rich integration of standards of MDA tools.

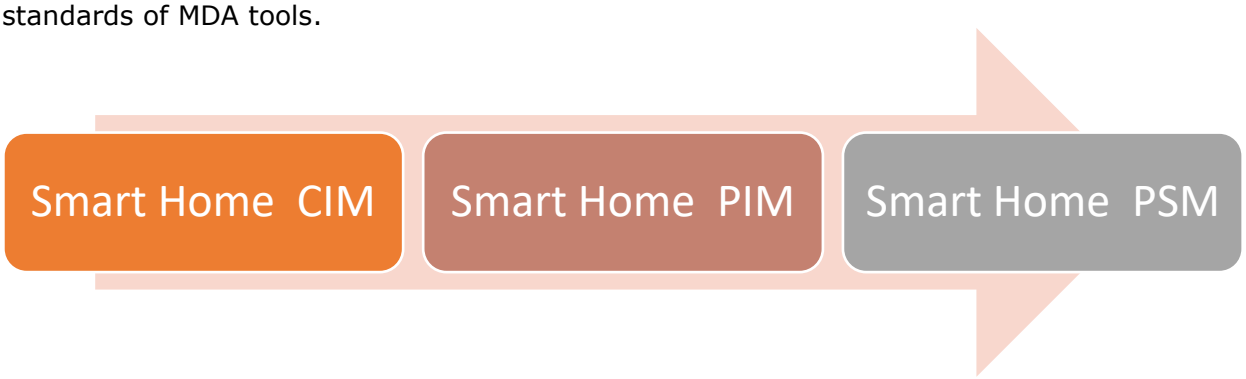

Figure 10. Smart Home SoS Model Transformations

As interoperability for Smart Home SoS is categorized in three different levels.

Interoperability Types for Smart Home SoS	RPC	RMI	XML	JSON	WSDL	SOAP	CORBA	MOF	CWM	UML Meta Model
Network Interoperability	Y	Y	Y	Y	Y	N	N	N	N	N
Syntactic Interoperability	N	N	Y	Y	N	Y	Y	N	N	N
Basic Connectivity Interoperability	N	N	Y	N	N	N	N	N	Y	Y

Figure 11. Smart Home SoS Interoperability Types

Following graph shows the roles of MDA Components

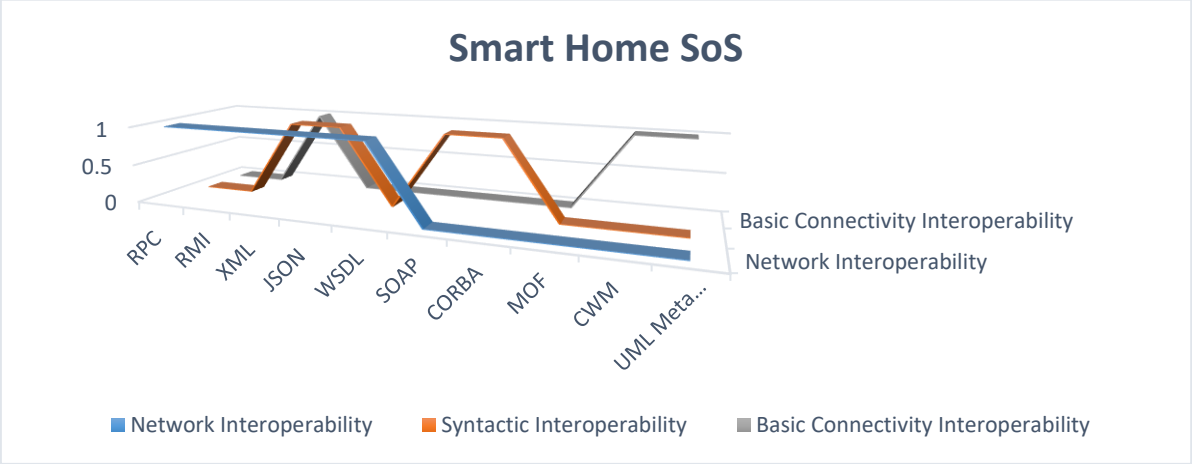

Figure 12. Smart Home SoS Interoperability Types graph

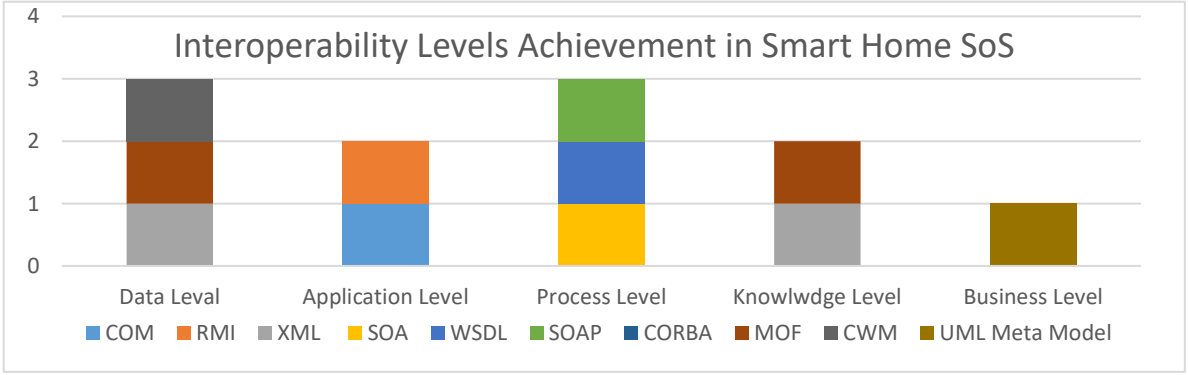

Figure 13. Smart Home SoS Interoperability level

5. Conclusion & Future Work

The research was carried out to explore the idea that how to meet with interoperability issues accumulated in the field of emerging complex heterogeneous System of Systems software development of the current era. With the dawn of technology humans rushed towards immense oceans of complications that leads towards the uncertainty for not only developers but all stakeholders. The new technology revolutionized the humanity by globalizing it. This evolving situation causes a huge diversity among systems and thus increase the risk of interoperability issues among collaborative systems.

5.1. Conclusion

The second question was "Why Interoperability issues occur in SoS implementation?" and we found that with the passage of time the complexity increases among software and in the past few years massive fields bloom in the oceans of computer science i.e. "Internet of Things" (IOT) for big data issues and "System of Systems" (SoS) as most complex heterogeneous systems collaborative work. Increase in heterogeneity and data persuaded are the reasons that cause interoperability issues among SoS.

We elaborated the role of architecture for systems that architecture is a basic structure that provides base to design anything. So software architecture, simply we can say that the conceptual structure of software elements.

As there are different architectural style available now a days. So a questions arises in our minds, "Why MDA while we've many other architecture styles?" we know that SoS are large and complex heterogeneous systems. Keeping in mind, there is the need of some standards to meet with the broad heterogeneity issues. After some research it reveals that an organization named "Object Management Group" (OMG) claims that there new standards of "Model Driven Architecture" (MDA) are enough to meet with the interoperability issues and study reveals that there claim is quite reasonable.

The determination of this research is to "Evaluate the Model Driven Architecture (MDA) for System of Systems (SoS) Interoperability". A case study of "Home Automation" as "Smart Home" is defined as it satisfied all the terms and conditions that put the smart home system in the line of SoS. MDA base models designed to evaluate the interoperability issues. Platform Independent Model (PIM) designed using UML. A tool named "Enterprise Architect" by Sparx Systems is selected to implement MDA. The platform-independent model is successively transformed to many platform-specific models (PSM) by mapping the PIM to some operation language or platform (e.g., Java) using prescribed rules.

PIM provides high level of abstraction and can easily be transformed to different heterogeneous platforms PSMs with the help of standards like XML, XMI, MOF, CORBA, CWM, WSDL, SOAP, Meta Models and COM etc. MDA also provides transformation standards among PSM to PSM and also the PSM to Code Generation. MDA not only used to design models but also provide the facility of design once implement many. It I observed that Model Driven Architecture (MDA) encapsulates all the standards used as communication channel among different systems. Basic Communication Interoperability (BCI), Network Interoperability (NI), Syntactic Interoperability (SI) are three different types of interoperability selected for smooth work of System od Systems (SoS) among many levels proposed in past researches and evaluation shows that the rich standards and abstraction provided by MDA can handle these types of interoperability by collaborative working.

Model Driven Architecture (MDA) provides reasonable standards to overcome System of Systems (SoS) Interoperability issue by not only providing an abstract Platform Independent Model (PIM) using some standard tools like Unified Model Language (UML) but also define rules to transform this abstract PIM to multiple specialized Platform Specific Model (PSM) and PSM to PSM transformation leading towards code generation using code engineering transformation rules.

5.2. Future Work

We spent many years on this research and still there many more things need to explore about this area but due to shortage of time we can't do so.

5.2.1. Short Term Objectives

Now humans are moving towards big data and to handle large complex problems still we need more research. PIM to PSM transformation is mature enough to compete System of Systems (SoS) collaboration issues to interoperate in heterogeneous environment but the code engineering still needs some improvements. There are more other tools still need to explore.

5.2.2. Long Term Objectives

Designing an MDA base architectures is moving towards more complex. "A significant step in the direction of an engineering discipline of software is a formal basis for describing and scrutinizing these designs. A formal approach to one aspect of architectural design: the interactions among components can be used. The key idea is to define architectural connectors as explicit semantic entities. These are specified as a collection of protocols that characterize each of the participant roles in an interaction and how these roles interact. To use this scheme a variety of common architectural connectors are defined. A formal semantics and show how this leads to a system in which architectural compatibility can be checked in a way analogous to type-checking in programming languages". (Allen & Garlan, 1997)

As Formalism is an emerging field in modern software engineering so. In future we also have a vision to apply formalism in designing Model Architecture for System of Systems Interoperability.

References

Allen, R., & Garlan, D. (1997). A formal basis for architectural connection. *ACM Transactions on Software Engineering and Methodology (TOSEM), 6*(3), 213-249.

Blair, G. S., Bennaceur, A., Georgantas, N., Grace, P., Issarny, V., Nundloll, V., & Paolucci, M. (2011). *The role of ontologies in emergent middleware: Supporting interoperability in complex distributed systems.* Paper presented at the ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing.

Boardman, J., & Sauser, B. (2006). *System of Systems-the meaning of of.* Paper presented at the 2006 IEEE/SMC International Conference on System of Systems Engineering.

Clements, P., Garlan, D., Little, R., Nord, R., & Stafford, J. (2003). *Documenting software architectures: views and beyond.* Paper presented at the 25th International Conference on Software Engineering, 2003. Proceedings.

DeLaurentis, D. (2007). *Role of humans in complexity of a system-of-systems.* Paper presented at the International Conference on Digital Human Modeling.

Gervais, M.-P. (2002). *Towards an MDA-oriented methodology.* Paper presented at the Proceedings 26th Annual International Computer Software and Applications.

Gorod, A., Sauser, B., & Boardman, J. (2008). System-of-systems engineering management: A review of modern history and a path forward. *IEEE Systems Journal, 2*(4), 484-499.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering: The Journal of the International Council on Systems Engineering, 1*(4), 267-284.

Morris, E., Levine, L., Meyers, C., Place, P., & Plakosh, D. (2004). *System of Systems Interoperability (SOSI)*. Retrieved from

Perumal, T., Ramli, A. R., Leong, C. Y., Samsudin, K., & Mansor, S. (2010). Interoperability among heterogeneous systems in smart home environment. In *Web-Based Information Technologies and Distributed Systems* (pp. 141-157): Springer.

Popper, S. W., Bankes, S. C., Callaway, R., & DeLaurentis, D. (2004). System of systems symposium: Report on a summer conversation. *Potomac Institute for Policy Studies, Arlington, VA, 320*.

Siegel, J. (2005). *Why use the model driven architecture to design and build distributed applications?* Paper presented at the Proceedings of the 27th international conference on Software engineering.

Zhang, W., Mei, H., Zhao, H., & Yang, J. (2005). *Transformation from CIM to PIM: A feature-oriented component-based approach.* Paper presented at the International Conference on Model Driven Engineering Languages and Systems.