

“MBA” An Approach for Constructing Systems of Systems

Ms. PADMAVATHY.V

B.TECH (Information Technology), Avvaiyar College of Engineering and Technology for Women.

Abstract - Despite the fact that the idea of System of Systems (SoS) has turned out to be very well known, most applications are still hand created. In this paper we display a structure, called MBA for Memory-Broker-Agent, tending to the improvement of frameworks of frameworks from a designing point of view. The principle highlights of the structure result from the experience picked up from building a SoS for creating programming cooperatively. In the paper we review the prerequisites for building a SoS and show how they can be met by utilizing a multi-specialist substrate. The MBA structure is an initial move towards proposing a nonspecific stage for creating frameworks of frameworks.

Key Words: *Systems of Systems, Architecture, Multi-specialist framework, Knowledge administration, User interface*

1. INTRODUCTION

Since the principal presentation of the General System Theory (GST) by Ludwig von Bertalanffy in the 40s [1] various complex frameworks have been created in different areas. Additionally, the need of building frameworks effectively has prompted the rise of the field of frameworks designing and the formation of the International Council on Systems Engineering (INCOSE).

Today mind boggling applications require to let a few frameworks that have been produced freely cooperate, prompting the idea of arrangement of frameworks (SoS). The thought of SoS from a frameworks building point of view, concentrated on interfacing free frameworks together, picked up force with the United States Strategic Defense Initiative (SDI) from the late 1980s [2].

These days, SoS has been connected and created in various distinctive areas, e.g., to enhance the nature of the code [3], to evaluate the manageability of various wellsprings of vitality [4], to help producing [5], to enhance understanding medicine solutions [6], to help tourism [7], to help military exercises ([8]; [9]; [10]), or to help creating atom smasher offices [11].

Be that as it may, most SoS are still hand created and no accord has yet been come to about an exact meaning of what they are ([12]; [13]; [14]; [15]; [16]). In this paper, we show a structure to encourage the advancement of frameworks of frameworks from a building point of view. We decide basic components and talk about how the relating engineering answers the vast majority of the prerequisites for building frameworks of frameworks.

2. REQUIREMENTS FOR BUILDING AN SOS

The absence of agreement in the meaning of frameworks of frameworks has lead analysts to begin constructing their definitions in light of SoS qualities, which as per Maier [17] are the accompanying.

- Operational Independence: all the constituent frameworks of a SoS can regularly convey their functionalities when not working with different constituents.
- Managerial Independence: every constituent arrangement of a SoS is administered by its own particular guidelines as opposed to by others outer to the constituent.
- Evolutionary Development: capacities and reasons for a SoS can progressively change and constituent frameworks can be added or evacuated to fit them.
- Emergent Behavior: a SoS is fit for conveying new functionalities that are the consequence of the constituent frameworks cooperating.
- Geographic Distribution: constituent frameworks of a SoS are topographically conveyed, implying that they can promptly trade just data and not significant amounts of mass or vitality.

Note that such highlights are additionally referred to by the INCOSE [18] as valuable to characterize what a SoS can be.

As a result of such attributes, building up a SoS addresses a few difficulties like interoperability, strength, information administration, client inclusion, or development.

a) Interoperability: In a SoS, constituent frameworks need to trade data and collaborate. Interoperability can be characterized as the capacity of particular frameworks to share semantically good data and after that procedure and oversee such data in semantically perfect ways, empowering clients to perform wanted errands (Zeigler et al. [19]; Madni and Sievers [20]). Giving interoperability between constituent frameworks that don't interoperate locally can require generous exertion and cost. To help SoS interoperability, correspondence conventions should be utilized, disentangling and dealing with the associations among heterogeneous frameworks.

b) Robustness: Because of the transformative idea of SoS, interdependencies and relations between SoS constituents can be altered, implying that progressions can happen in an unpredictable way (De Laurentis [21]). In this way, it is foremost to think about strength in SoS. With regards to SoS,

strength can be characterized as the capacity to convey ability in obscure future conditions.

c) Knowledge administration: Performing Knowledge Management (KM) is basic as SoS change after some time. Since frameworks and clients may change amid SoS life cycle, vital information must be gathered, sorted out and saved.

d) User measurement: Besides being an imperative single characteristic, the client measurement is likewise a test in SoS. Dissimilar to in solid frameworks in which clients have predefined interfaces, in a SoS clients cooperate through evolving between faces with different frameworks or with clients of the SoS (Madni and Sievers [20]). Such a dynamic conduct happens fundamentally on the grounds that frameworks can be included, evacuated or supplanted in a SoS. Additionally, collaborations can happen through interfaces in outside frameworks associated with the SoS, for example, a Web server or another application. Moreover, data for basic leadership ought to be given in the interest of people, i.e., should take a client point of view, giving redid data maybe through AI-based frameworks.

e) Evolution: Evolution implies changing a SoS, for example, by including, evacuating, supplanting or altering its constituent frameworks. As indicated by Agarwal et al. [22], a portion of the requirements for developing a SoS can be for:

- (i) enhancing SoS execution;
- (ii) rendering constituent frameworks interoperable;
- (iii) including extra prerequisites for the SoS objective;
- (iv) taking care of development in the constituent frameworks; and
- (v) adding new functionalities to the constituent frameworks. A standard practice prescribed to deal with the development in SoS is to leave constituent frameworks approximately coupled.

The accompanying segment proposes another approach, conveying answers to the difficulties we just said.

3. THE MBA ARCHITECTURE

Thinking about the attributes and difficulties for building frameworks of frameworks, one can't yet consider multi-operators frameworks (MAS). In reality, despite the fact that a specialist in a MAS more often than not can't work genuinely without the help of alternate operators, numerous MAS highlights constitute great contender for supporting a SoS engineering. The approach hence comprises of building frameworks of frameworks over a multi-specialist layer that will give the required systems to noting the prerequisites. Such an approach is propelled by the PACT (Palo Alto Collaborative Testbed) venture (Cutkosky et al. [23]) that

presented the possibility of facilitator and enabled Gruber to build up the idea of cosmology.

3.1 Generally Approach

The principle thought is to render the part frameworks interoperable by utilizing facilitator operators, one for every framework. At that point, we connect the operators to merchants (Park et al. [24]) that will deal with sorting out trades utilizing a standard convention. The MAS stage will likewise offer the likelihood of adding Personal Assistant specialists to interface with clients, and operators responsible for recording data important for overseeing learning. Along these lines, the interoperability and vigor issues will be tended to by identifying the segment frameworks utilizing facilitators, handles, a standard convention and ontologies; information administration, by the presence of specialists accountable for memory; client measurement, by close to home partner operators; and development, by a plausibility of fitting and play of the segment frameworks. Thus, the name of the proposed system: MBA for Memory-Broker-Agents.

3.2 The MBA Framework

The MBA model can be characterized as a tuple $\psi := (M, B, A, Y)$, where

- An are limited arrangements of recollections, intermediaries, and operators, separately.
- Y is a ternary connection among them, i.e., $Y \subseteq M \times B \times A$. This connection determines the coupling among parts.

The MBA design is a space free center architecture, implying that it is reached out as indicated by the area, objectives and frameworks of the SoS being created. In the architecture, three primary components are recognized: "framework," "agent" and "memory." A negligible case of the proposed design is appeared in Fig. 1.

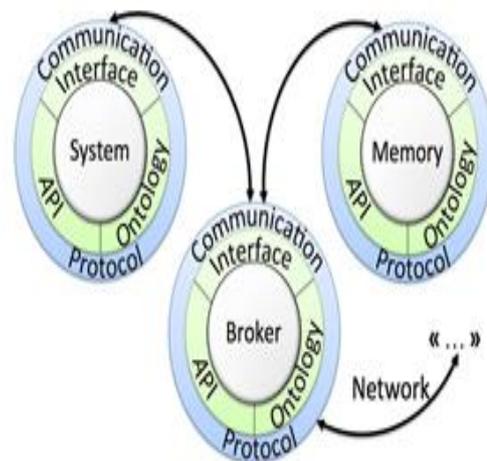


Fig. 1. The MBA design proposed in this exploration.

The framework component speaks to a part arrangement of a SoS. To end up a constituent arrangement of a MBA SoS, a system must give no less than an API (Application Programming Interface). Subsequent to being interfaced through a facilitator operator, the framework will be enlarged by a correspondence convention, ontologies, and discretionary interfaces.

The correspondence convention is utilized to trade messages and data between the framework and the other constituent frameworks in the MBA SoS.

The intermediary component is planned to get demands from segment frameworks, endeavor to discover potential suppliers, and afterward once the errands are allotted, exchange the outcomes back to the guests. Notwithstanding giving free coupling between frameworks, an essential normal for the specialist is that it tends to be a solitary operator or an arrangement of representative operators. Utilizing a few dealers can abstain from over-burdening a solitary agent, in this manner improving effectiveness, and making the MBA design more strong. Also agents utilize a Contract-Net convention contending administrations, choosing from the most productive one, or effortlessly eliminating a subsystem when another rendition shows up.

The memory component is there to underwrite and oversee information concerning the SoS and its space. In our approach, it is conceivable to have the same number of recollections as required or required by a given area, including excess components. The fundamental preferred standpoint of recollections in our approach is that we can reuse information from the SoS and its area. Recollections are typically outer frameworks. Presently, the MBA design gives facilitator operators prepared to interface contrast ent sorts of recollections in view of protest, social, key-esteem and triple store information models. The benefit of doing this, is our approach encourages crafted by SoS modellers, giving them a chance to modify such facilitator specialists for their application.

When the sum total of what components have been "identified," one must give a standard convention to asking for administrations and getting a swers and guarantee both syntactic and semantic interoperability.

3.3 The MBA Communication Protocol

The low level convention is given by the supporting multi-specialist stage. At a more elevated amount, demands for administrations are demonstrated utilizing work requests and answer arranges by answer designs.

1) Work Orders: A work arrange portrays a demand sent to an agent. Formally, it tends to be characterized as takes after:

`<work order> ::= (<message-content>[, <action>], <timeout>[, <C-net strategy>])`

where,

- `<message-content>` is the substance or contentions of the work arrange definite beneath.
- `<action>` is a discretionary parameter, being the name (a watchword) of an asked for usefulness. To prevail with regards to asking for `<action>`, the requester must seem some place in the metaphysics of the potential suppliers.
- `<timeout>` is a number speaking to the most extreme measure of time a requester will hold up to get a solution to its demand.
- `<C-net strategy>` is a parameter utilized by requesters to determine the methodology utilized in a Contract Net for finding solutions.

Since there might be diverse purposes behind a demand not to be replied, e.g. the facilitator operators interfacing the frameworks are completely bustling taking a shot at a few solicitations, or they basically would prefer not to reply, the MBA design receives the approach that frameworks are allowed not to answer demands. Thus, it is important to permit determining timeouts in the work arrange.

The contentions or substance of a work arrange, i.e., `<message-content>`, can be characterized as takes after:

`<message-content> = ([<query>], [<data>], [<pattern>], <language>)`

where,

`<query>` speaks to a demand made in an inquiry arrange (the ontologies of the facilitator operators of requesters and suppliers must be in any event perfect).

- `<data>` remains for some info information.
- `<pattern>` an answer design deciding how results ought to be organized.
- `<language>` the dialect utilized in the message content.

2) Answer Patterns: An answer design in the MBA architecture determines how the outcomes from a demand ought to be organized with a specific end goal to be comprehended by the requester. It is a rundown of etymological signals communicated as a tree of ontological terms. These terms are ideas and properties taken from the requester's cosmology. The example offers semantics to solicitations and results.

`<pattern> ::= (<concept> ({<attribute>}*))`

`| {<relation> <pattern>}*`

3.4 Ontologies

Ontologies constitute the foundation of our engineering. They assume four distinct parts:

(i) displaying the SoS area; (ii) demonstrating the clients of constituent frameworks; (iii) deciphering associations in normal dialect; (iv) dealing with the semantics of the correspondence convention.

Since a SoS is made out of frameworks that were assembled autonomously and on the grounds that its structure will change after some time, it isn't conceivable to build a solitary cosmology that could homogenize all its diverse parts. It is conceivable anyway to adjust parts of the distinctive ontologies inside the application to guarantee a negligible comprehension.

A second part of the ontologies is to demonstrate clients of the segment frameworks. Speaking to the conduct of the client communicating with a segment framework will help give customized support to enhancing her connection with the SoS. This should be possible regarding the client's close to home collaborator.

Ontologies are additionally fundamental for interpreting connections in normal dialect. On the framework side, the objective for this situation is to encourage disentangling, translating and understanding the articulations amongst clients and their PAs.

Ontologies are likewise important to translate articulations of the message content dialect, guaranteeing semantic interoperability.

The ontologies used to help the semantics are kept inside the facilitator operators. It is vital to feature that such ontologies can vary in every operator, being specific as indicated by the qualities of the frameworks they are interfacing. In this way, the ontologies of two given facilitator operators require not be the same but rather should have an insignificant level of similarity to permit trading data reasonably.

3.5 Condensing Process

Subsequent to achieving the assigned undertakings, suppliers are responsible for organizing the outcomes in an arrangement justifiable by the requester. To do that, they should play out an outlining procedure, which comprises of filling an answer design with the important data. Every supplier, through its facilitator, must adjust its cosmology to the substance of the appropriate response design in the work arrange, at that point sort out its information to structure it as per the example.

3.6 Client Dimension

SoS have various human clients who have two sorts of interfaces: an interface with a segment framework, and an interface with the SoS. Since point and snap interfaces are

not effective in complex conditions, and on the grounds that with the MAS substrate we can utilize individual associate specialists (PA), a characteristic method for connecting with the SoS is by utilizing regular dialect composed or talked.

A PA is an essential resource since it can act proactively, give modified help, decrease the client's intellectual load, help to expand collaboration with different SoS clients, and handle multimodal cooperation's. In addition it has been demonstrated that regular dialect discoursed in an expert setting are not extremely hard to execute (Barthes' [26], Fechner et al. [27]). At last, discourse to-content and content to-discourse programs have gained enough ground to be presently incorporated in such sorts of interfaces (Jones et al. [28]).

4. CASE STUDY

We constructed a model in the area of programming improvement to survey the distinctive issues that could emerge.

4.1 Review

The target of programming improvement is to help colleagues working cooperatively to improve the nature of the code they create in a given venture. The motivation behind the focused on SoS was the accompanying: While designers are composing their code, learning about the code quality is promoted naturally and in a non meddling manner; at that point, input is given demonstrating conceivable quality issues and suggestions for tending to them; in parallel, directors get data about the nature of undertakings or the nature of the code through outlines or tables, to enable them to decide.

The SoS appeared in Fig. 2 comprises of coding frameworks, a code examination framework, a forming framework, various types of Databases (DB, for example, social, protest, key-esteem, and triple store, MBA Browsers frameworks, and a Web look framework. Every one of the frameworks are operational and administrative free, however participate together planning to enhance the nature of the code.

The general SoS fills in as takes after. Engineers utilize their IDE to create (Java) code. They can compose code of a given task that has its quality considered by the SoS, or to create code in parallel identified with different ventures which are not considered by the SoS. The created code is kept in a forming framework which can store code progressively from various diverse activities. The investigation framework recovers the code from the forming framework, performing quality examination. In the meantime the examination framework collaborates with the SoS, it can likewise give investigation to code from various groups and undertakings.

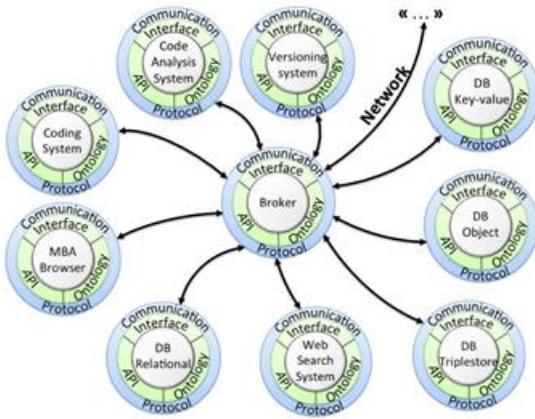


Fig. 2. The proposed MBA design connected to the setting of community programming advancement.

At the point when shortcomings are discovered, suggestions are removed from the different databases and from the Web which may likewise work in parallel for putting away or recovering data outside the SoS. From that point onward, the proposals are sent to the designer through her coding framework. An intelligent vocal interface permits speaking with the framework for asking for administrations or with different engineers to get data. The task supervisor has an outside framework with a Web interface taken care of by a Personal Assistant, enabling her to ask for and get data from the SoS, for example, to see the advancement of the improvements and the issues experienced.

4.2 Execution

The coding framework utilized by every designer was Eclipse1. The forming framework was github2. The investigation framework was a multi-operator framework applying principles to figure the nature of the produces Java code. The social database was MySQL3, the question database was AllegroCache4, the key-esteem database was Redis5, the triple store was really an information administration framework dealing with records saw as assets MEMORAE6 (Abel [29]). The supporting MAS platform was OMAS7 (Barthes` [30]). Facilitators were occurrences of the OMAS Transfer Agents, interfaces were executed as cases of OMAS Personal Assistant Agents. The Broker was an occasion of OMAS Service Agent. All Web interfaces were utilized the likelihood of survey the MAS stage as a Web server. Communications utilized the stage Contract Net office. Vocal interface was executed utilizing discourse to-content and content to-discourse programming.

4.3 Talk

Building the model of SoS for creating programming cooperatively enabled us to decide the critical component a non specific stage should offer. The part of the MAS amid usage was essential since it gave operator models to building the facilitators, the individual collaborator specialists for between confronting people and the Contract-Net convention for trading messages. The OMAS condition

additionally offered help for creating ontologies effortlessly and a thinking procedure like Jena or SPARQL. Work orders were worked in a paltry manner, and OMAS offered the outlining component to give facilitators a chance to organize their answers. Normal dialect preparing and exchange instrument were likewise bolstered by existing structures. A troublesome indicate anyway was assemble the real close to home aide exchanges, which required a long and monotonous work. Building facilitators utilized the accessible Transfer Agent component, however required to introduce a particular philosophy and interpreters for coordinating the between specialist message structure to the mannerisms of every component framework, for instance, coordinating the metaphysics with social tables for the MySQL facilitator.

A vital revelation was that we could supplant the merchant operator by utilizing restrictive tending to permitting to convey messages just to specialists that fulfill certain conditions detailed utilizing the sender cosmology and the metaphysics question structure. The OMAS stage is very productive with communicated messages or Contract-Net messages since the call for offers should be possible with a solitary message, and in addition conceding an errand. This approach, characterizing a virtual specialist, safeguards the P2P idea of the trades among operators.

5. Worldwide DISCUSSION

Subsequent to showing the approach, let us see now how it answers a portion of the difficulties recorded toward the start of the paper.

- Interoperability is given by the fundamental multi-specialist stage and the utilization of facilitators to embed outside frameworks or inheritance frameworks. The MAS stage likewise permits topographical conveyance.
- Robustness is enhanced by the utilization of Contract-Net, which permits embedding new parts in parallel to existing ones and improving administrations.
- Knowledge administration must be executed in the structure of every application except can depend on the interface of databases (social, question, or other) or more advanced frameworks like MEMORY
- User measurement is considered by the likelihood of interfacing clients to the SoS through Personal Assistant specialists, directing trades utilizing normal dialect in a printed or vocal mode. Individual collaborators can be proactive and make utilization of client profiles to more readily modify connections.
- Evolution is supported by the free coupling of component frameworks and the utilization of standard Work Orders and Answer Patterns in the convention. Free coupling guarantees the operation rational and administrative autonomy of the part frameworks.

Note that the proposed structure does not make a difference to every one of the attributes of an arrangement of frameworks, for instance concerning rise. Such qualities identify with applications. However, by endeavoring to answer challenges, we trust that the MBA structure will prompt create frameworks of frameworks showing the attributes said toward the start of the paper.

Concerning related work, there isn't sufficient room here to say the various papers that have talked about the engineering of a SoS. We look at some ongoing ones.

Numerous methodologies depend on brought together segments, for in-position, to store and give the administrations or functionalities utilized by frameworks in the SoS or to render constituent frameworks interoperable concentrating on semantics and interpretations (e.g. Perez et al. [31] or Varga et al. [32]). The utilization of a concentrated part makes the engineering more fragile, making single purposes of disappointment, which is much more hazardous on account of SoS in light of rising and erratic conduct. Besides, the focal part can be over-burden with demands, therefore the SoS can experience the ill effects of bottlenecks, which can influence the normal outcomes. Also, the utilization of focal components prompts tight coupling, which isn't proper for adapting to the development of the framework.

As to, most methodologies utilize or recom-repair the utilization of norms. A few works like (Wong et al. [33]) utilize a few conventions or guidelines in the meantime. For this situation, message interpretation is done inside focal parts. This expands the many-sided quality, exertion and cost, and each time a framework utilizing an alternate convention is associated with the SoS, new interpretation instruments should be made and added to the focal component. Now and again, keeping in mind the end goal to actualize the progressions, the focal component should be quit, conveying the whole SoS to a stop.

A few works take a more reasonable view with respect to architectural bolster for SoS. A few creators propose dialects to formally depict the engineering (Oquendo [34]), or strategies to streamline and give basic leadership utilizing SoS models (Agarwal et al. [22]). Others attempt to think about engineering designs (Ingram et al. [35]). Despite everything others, similar to Ge et al. [36] propose utilizing surely understood engineering structures, for example, DoDAF. Be that as it may, such structures speak to the engineering statically and concentrate excessively on what ought to be portrayed instead of on the pragmatic issues. They center for the most part around a theoretical level, i.e., depicting and recording the structures, as opposed to a down to earth perspective for supporting the advancement of SoS.

As to measurement, most methodologies utilize the WIMP worldview (Windows, symbols, menus, pointer), hard to use on account of SoS as data is traded among various types of frameworks. Besides, the utilization of inflexible interfaces

can't be suitable for SoS as they change after some time. A few creators offer cooperation in common dialect, however on a restricted scale.

In spite of the fact that heartiness is an imperative issue, we didn't discover an approach that endeavors to help SoS vigor from a viable perspective, i.e., straightforwardly amid SoS activity. For example, none of the methodologies endeavor to convey back a SoS to an intelligent state after its constituent frameworks have gone down.

This can be critical since once frameworks are reconnected to the SoS their state may never again be reliable with whatever is left of the SoS, and hence their data can be ambiguous and spread over every one of the SoS.

Throughout the years, a developing number of creators have been attempting to arrange SoS, for example, to control the choice of architecting standards. Four primary classes (Maier [17]; Dahmann and Baldwin [37]) in light of the specialist connection delivers between the SoS and the constituent frameworks have been broadly embraced (Dahmann and Roedler [38]):

- **Directed:** In these SoS, the constituent frameworks are subordinated to a focal expert to satisfy a particular reason. The constituent frameworks of the SoS can work autonomously, however are figured out how to fulfill the objective reason.
- **Collaborative:** The constituent frameworks interface and collaborate deliberately to satisfy the concurred basic reason. In this classification of SoS, a focal administration organization does not have coercive capacity to run the framework.
- **Acknowledged:** It is a half breed of the coordinated and collaborative SoS. There is an administration expert at both the SoS and the framework levels. The recognized SoS have clear purposes, administration and assets. The constituent frameworks proceed as autonomous substances, seeking after their own particular objectives with free administration, assets, partners. There is a simultaneous overseement. Contending interests and needs may emerge.
- **Virtual:** In this sort of SoS, there is an absence of both focal administration specialist and midway settled upon purposes. Substantial scale conduct develops, and might be alluring, yet the SoS must depend upon generally invisible components to look after it. A virtual SoS might be think or incidental.

Presently, how could the MBA system help to construct such sorts of SoS?

In the first place, take note of that the constituent frameworks of a SoS worked through our structure are not normally con-stressed by any type of administrative control. The principle reason is on the grounds that we will probably give a bland approach, in this way we abstain from forcing

such an imperative. Nonetheless, we depict now how our structure could be utilized for building SoS of the four sorts specified previously.

Coordinated SoS: when constructing a coordinated SoS with the MBA structure, the focal specialist could utilize outside frameworks with Personal Assistant (PA) operators to control the SoS by asking for or getting its functionalities. Our approach as of now gives such an outside framework which is the MBA Browser utilized by programming directors for the situation investigation of Section III. The entrance to the SoS functionalities by both, the focal specialist and constituent frameworks, could be modified in facilitators of constituents and exchanges of the PAs.

Cooperative SoS: for building a shared SoS with our approach, a draftsman may utilize the MBA system directly. For example, she needs to interface potential constituent frameworks with facilitators and connection them with merchants, as de-scribed in this paper. The explanation behind that is on the grounds that intrinsically a SoS worked through the MBA structure isn't constrained to focal specialists. A case of shared framework fabricated utilizing our approach can the contextual analysis of collective programming advancement depicted in Section III.

Recognized SoS: in a recognized SoS, the focal expert could likewise utilize outer frameworks connecting through PAs, as we have prescribed for the Directed SoS compose. When managing contending interests and needs, constituent frameworks could depend on a more specialized level on the Work Order convention, by utilizing timeouts and parametrizing facilitators to answer or not demands. In any case, if a more refined approach is required, at that point it must be executed by the modeler, for example, indicating in the facilitator specialists how to deal with the approaching Work Orders.

Virtual SoS: for building a virtual SoS with the MBA structure, first the designer would be required to interface potential constituent frameworks with facilitators, not surprisingly. At that point, on the grounds that a virtual SoS does not have a focal specialist and not by any means a particular reason, the facilitators could be tweaked for permitting the constituents ask for and get maybe all functionalities from each other uninhibitedly.

Following this line of arranging SoS through sorts, the MBA SoS we worked for the situation investigation of Section III could be considered of the cooperative kind. The fundamental reasons are that, first the constituent frameworks don't depend on focal experts controlling the SoS. That is, the connections between them are performed for the most part as indicated by the SoS objective of enhancing code quality. The partners, for example, programming improvement supervisors and engineers are allowed to ask for the SoS functionalities, in consonance with their parts in the SoS space. In addition, in the application level, supervisors and engineers can team up, for example,

by proposing suggestions to enhance the code quality. Be that as it may, engineers are allowed to acknowledge or not such recommendations.

The fundamental contrasts between crafted by the writing to the approach proposed by this exploration is that we propose a space autonomous shared (P2P) design with free coupling between its components, concentrating on the advancement of a SoS from a functional perspective. Our approach does not depend on brought together parts or keep references between every it components. Also, on account of the free coupling given by our design, the SoS created with it very well may be effectively adjusted to various spaces. Besides, our approach gives fundamental components to information capitalization and man-agement, and utilizations a solitary correspondence convention giving syntactic and semantic interoperability between constituent frameworks. It likewise considers the client measurement by furnishing proactive interfaces equipped for cooperating with SoS clients through discoursed with Personal Assistants, and offering proactive help when they are utilizing the SoS. Also, our approach thinks about power amid SOS task, attempting to help it keeping rational when constituent frameworks go down.

The work done as of not long ago enabled us to propose the MBA structure. We are at present building up a nonspecific stage utilizing this structure, and in addition a technique for building frameworks of frameworks. We began testing the approach on new issues, in particular in the domain of health care (Wanderley et al.[6]).

REFERENCES

- 1) L. v. Bertalanffy, General system theory: Foundations, development, applications. George Braziller, 1969.
- 2) C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Pe-leska, "Systems of systems engineering: basic concepts, model-based techniques, and research directions," ACM Computing Surveys (CSUR), vol. 48, no. 2, p. 18, 2015.
- 3) G. M. P. Wanderley, M.-H. Abel, J.-P. Barthes, and E. C. Paraiso, "A system of systems architecture for supporting decision-making," in Computer Supported Cooperative Work in Design (CSCWD), 2017 IEEE 21st International Conference on. IEEE, 2017, pp. 186–191.
- 4) S. Hadian and K. Madani, "A system of systems approach to energy sustainability assessment: Are all renewables really green?" Ecological Indicators, vol. 52, pp. 194–206, 2015.
- 5) S. Ford, U. Rauschecker, and N. Athanassopoulou, "System-of-system approaches and challenges for multi-site manufacturing," in System of Systems

- Engineering (SoSE), 2012 7th International Conference on. IEEE, 2012, pp. 1–6.
- 6) G. M. P. Wanderley, E. Vandenberg, M.-H. Abel, J.-P. A. Barthes, M. Hainselin, H. Mouras, A. Lenglet, M. Tir, and L. Heurley, "Con-signela: A multidisciplinary patient-centered project to improve drug prescription comprehension and execution in elderly people and parkinsonian patients," *Telematics and Informatics*, 2017.
 - 7) L. Bai, "System of systems engineering and geographical simulation: Towards a smart tourism industry information system," in *Advanced Communication Technology (ICACT)*, 2013 15th International Conference on. IEEE, 2013, pp. 1015–1018.
 - 8) P. Hershey, M.-C. Wang, and D. Toppin, "System of systems for autonomous mission decisions," in *System of Systems Engineering (SoSE)*, 2013 8th International Conference on. IEEE, 2013, pp. 129–134.
 - 9) J. P. Olivier, S. Balestrini-Robinson, and S. Briceno, "Approach to capability-based system-of-systems framework in support of naval ship design," in *Systems Conference (SysCon)*, 2014 8th Annual IEEE. IEEE, 2014, pp. 388–395.
 - 10) J. C. Kilian and T. M. Schuck, "Architecture and system-of-systems design for integrated missile defense," in *System of Systems Engineering Conference (SoSE)*, 2016 11th. IEEE, 2016, pp. 1–6.
 - 11) T. Friedrich, C. Hilbes, and A. Nordt, "Systems of systems engineering for particle accelerator based research facilities: A case study on engineering machine protection," in *Systems Conference (SysCon)*, 2017 Annual IEEE International. IEEE, 2017, pp. 1–8.
 - 12) M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4<267::AID-SYS3>3.0.CO;2-D](http://dx.doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D)
 - 13) R. L. Ackoff, "Towards a system of systems concepts," *Management science*, vol. 17, no. 11, pp. 661–671, 1971.
 - 14) J. P. Dauby and S. Upholzer, "Exploring behavioral dynamics in systems of systems," *Procedia Computer Science*, vol. 6, pp. 34–39, 2011.
 - 15) C. Sary and D. Wachholder, "System-of-systems supporta bigraph approach to interoperability and emergent behavior," *Data & Knowledge Engineering*, vol. 105, pp. 155–172, 2016.
 - 16) M. Jamshidi, *Systems of systems engineering: principles and applications*. CRC press, 2017.
 - 17) M. W. Maier, "Architecting principles for systems-of-systems," *INCOSE International Symposium*, vol. 6, no. 1, pp. 565–573, 1996. [Online]. Available: <http://dx.doi.org/10.1002/j.2334-5837.1996.tb02054.x>
 - 18) D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin, and T. M. Shortell, *Systems engineering handbook: A guide for system life cycle processes and activities*. John Wiley & Sons, 2015.
 - 19) B. P. Zeigler, S. Mittal, and X. Hu, "Towards a formal standard for interoperability in m&s/system of systems integration," in *GMU-AFCEA Symposium on Critical Issues in C4I*, 2008.
 - 20) M. Madni and M. Sievers, "System of systems integration: key systems of systems design methods," in *Systems, Man and Cybernetics*, 2005 IEEE International Conference on, vol. 1. IEEE, 2005, pp. 86–91.
 - 21) S. Agarwal, L. E. Pape, C. H. Dagli, N. K. Ergin, D. Enke, A. Gosavi,
 - 22) Qin, D. Konur, R. Wang, and R. D. Gottapu, "Flexible and intelligent learning architectures for sos (filasos): Architectural evolution in systems-of-systems," *Procedia Computer Science*, vol. 44, pp. 76–85, 2015.
 - 23) M. R. Cutkosky, R. S. Engelmores, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber, "Pact: an experiment in integrating concurrent engineering systems," *Computer*, vol. 26, no. 1, pp. 28–37, Jan 1993.
 - 24) H. Park, J. Tenenbaum, and R. Dove, "Agile infrastructure for manufacturing systems: A vision for transforming the us manufacturing base," in *Proceedings of the Defense Manufacturing Conference*, 1993.
 - 25) R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on computers*, no. 12, pp. 1104–1113, 1980.
 - 26) J.-P. A. Barthes, "Flexible communication based on linguistic and ontological cues," in *E-Technologies: Transformation in a Connected World - 5th International Conference, MCETECH 2011, Les Diablerets, Switzerland, January 23-26, 2011, Revised Selected Papers*, 2011, pp. 131–145. [Online]. Available: https://doi.org/10.1007/978-3-642-20862-1_9

- 27) M. Fuckner, J.-P. A. Barthes,` and E. E. Scalabrin, "Using personal assistant dialogs for automatic web service discovery and execution," in WEBIST 2013 - Proceedings of the 9th International Conference on Web Information Systems and Technologies, Aachen, Germany, 8-10 May, 2013, 2013, pp. 189–198.
- 28) Jones, A. Kendira, C. Moulin, J.-P. A. Barthes,` D. Lenne, and Gidel, "Vocal interaction in collocated cooperative design," in Cognitive Informatics & Cognitive Computing (ICCI* CC), 2012 IEEE 11th International Conference on. IEEE, 2012, pp. 246–252.
- 29) M.-H. Abel, "Knowledge map-based web platform to facilitate organi-zational learning return of experiences," *Computers in Human Behavior*, vol. 51, pp. 960–966, 2015.
- 30) J.-P. A. Barthes,` "Omas a flexible multi-agent environment for cscwd," *Future Generation Computer Systems*, vol. 27, no. 1, pp. 78–87, 2011.
- 31) J. Perez,` J. D'iaz, J. Garbajosa, A. Yague,` E. Gonzalez, and M. Lopez-Perea, "Towards a reference architecture for large-scale smart grids system of systems," in *Proceedings of the Third International Workshop on Software Engineering for Systems-of-Systems*. IEEE Press, 2015, pp. 5–11.
- 32) P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, Delsing, and I. M. de Soria, "Making system of systems interoperable– the core components of the arrowhead framework," *Journal of Network and Computer Applications*, vol. 81, pp. 85–95, 2017.
- 33) R. K. Wong, C. H. Chi, Z. Yu, and Y. Zhao, "A system of systems service design for social media analytics," in *Services Computing (SCC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 789–796.
- 34) F. Oquendo, "Formally describing the software architecture of systems-of-systems with sosadl," in *System of Systems Engineering Conference (SoSE), 2016 11th. IEEE, 2016*, pp. 1–6.
- 35) C. Ingram, R. Payne, and J. Fitzgerald, "Architectural modelling patterns for systems of systems," in *INCOSE International Symposium*, vol. 25, no. 1. Wiley Online Library, 2015, pp. 1177–1192.
- 36) B. Ge, K. W. Hipel, K. Yang, and Y. Chen, "A novel executable modeling approach for system-of-systems architecture," *IEEE Systems Journal*, vol. 8, no. 1, pp. 4–13, 2014.
- 37) J. S. Dahmann and K. J. Baldwin, "Understanding the current state of us defense systems of systems and the implications for systems engineering," in *Systems Conference, 2008 2nd Annual IEEE. IEEE, 2008*, pp. 1–7.
- 38) J. Dahmann and G. Roedler, "Moving towards standardization for system of systems engineering," in *System of Systems Engineering Conference (SoSE), 2016 11th. IEEE, 2016*, pp. 1–6.

AUTHOR



Padmavathy V
B.Tech(IT),
Avvaiyar college of engineering
and technology for women