2024 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA) | 979-8-3503-6281-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CogSIMA61085.2024.10553991

Service Oriented Architecture For Knowledge Acquisition and Aggregation

Kenneth J. Hintz
Mechanical and Aerospace Engineering Dept.
State University of NY at Buffalo
Buffalo, NY, USA
khintz2@buffalo.edu

James Llinas
Industrial and Systems Engineering Dept.
State University of NY at Buffalo
Buffalo, NY, USA
llinas@buffalo.edu

Kari Sentz
Computer, Computational, and Statistical Sciences Division
Los Alamos National Laboratory
Los Alamos, NM, USA
ksentz@lanl.gov

***Abstract*—Situation management involves the acquisition and dissemination of knowledge to effect situation awareness and assessment. In a companion paper, the concept of Knowledge Acquisition, Representation, Processing, & Presentation (KARPP) system has been introduced. In that paper, it was proposed to manage an adaptive data fusion process including the decomposition of a sensor and database knowledge acquisition and aggregation system into a service oriented architecture (SOA) of knowledge aggregation services (KAS) with standards-based inputs and outputs which are implemented in hidden processes. This paper reviews current approaches to SOA and expands the KARPP architecture into a taxonomy of required and support services as well as discusses particulars of a self-organizing implementation of KARPP in a dynamic SOA of microservices. In addition to a taxonomy of services, a self-documenting approach is introduced to allow for preplanned product improvement (PPI) and incremental development as well as the concept of a Knowledge Aggregation System Service broker.**

Keywords—KARPP, knowledge aggregation, service oriented architecture, service broker

I. Introduction

The concept of Knowledge Acquisition, Representation, Processing, & Presentation (KARPP) for situation assessment and management has been presented in a companion paper [1] in which Knowledge Aggregation Services (KAS) were introduced as a fundamental unit of service for a self-similar hybrid implementation. A simple block diagram of this service aggregation showing the service requestor and service provider is shown in Figure 1. This recognizes that knowledge acquisition can be decomposed into a discretized and refined process progressing from the highest level of decision making and qualified knowledge queries to the lowest level of satisficing knowledge acquisition in order to implement an Observe, Orient, Decide and Act (OODA) loop as presented by Zhao et al. [2]. However, the resulting system cannot be a disjointed collection of stovepipe processes but needs to conform to an architectural concept that relates and integrates the various services in support of knowledge acquisition.

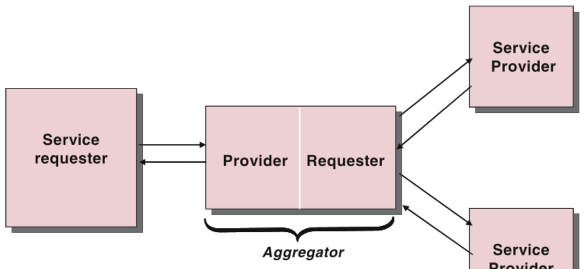

Figure 1 Role of a service aggregator and relationship of a requestor and provider [12].

A knowledge acquisition system for a decision maker can be reduced to an architecture of orthogonal services which can 1) determine what knowledge to acquire in order to make a decision, 2) the mission value of that knowledge, 3) the requisite quality associated with the knowledge, 4) the temporal needs for the knowledge, and 5) what is the best next means with which to acquire and process that knowledge. [1]

A Service Oriented Architecture (SOA) can meet that need if it is well-defined and extensible as has been shown in the model driven architecture for supporting decision making of Boumahdi [3].

Casola et al. [4] "...have developed a Sensor Web client capable of visualizing geospatial data, and a set of Web Services called GeoSWIFT. Actually, two main European early warning projects based on Service Oriented Architectures (SOA) have been proposed; the WIN (Wide Information Network) and ORCHESTRA projects. WIN aims at developing an open and flexible platform to support multi-hazard and risk domains at European level, integrating national and regional data flows in the frame of a Web Service Architecture. It proposes a set of generic services which are standard data modeling components that facilitate the deployment on various thematic cases. The WIN Metadata Model is based by and large on existing standards (such as the Dublin Core, GML, and ISO19115) and includes some additional specifications for WIN. On the other hand, the ORCHESTRA architecture adapts the ISO/IEC 10746 Reference Model for Open Distributed Processing to service-oriented architectures. In particular, The web services are implemented using W3C Web services platform and the Geography Mark-up Language (GML)."

Many knowledge acquisition systems are one of a kind, unique systems which are neither extensible nor dedicated to acquiring knowledge from a sensing system or from databases with the notable exception of some GIS spatial data systems by Tang et al. [5]. There is very little work on viewing both sensor data acquisition and data resulting from searching a database as essentially the same thing, knowledge. While all the components that we presented in KARPP (from knowledge presentation to the decision maker and the graphical user interface (GUI) input of qualified knowledge requests to the actual acquisition of the knowledge) are treated extensively and usually independently in the literature. In particular, data fusion and data source selection (whether a sensor or a database) are treated as distinct entities with no acknowledgement of the interdependencies between the two and the improvement that can be made in situation assessment by choosing the right combination of fusion method and data to fuse.

Section I of this paper briefly introduces the need and motivation for SOA as an implementation of KARPP. Section II reviews the state of the art of SOA particularly as it has been applied to military and government problem domains. Section III defines an extensible approach to describing individual KAS components of the SOA. Section IV introduces a partial listing of relevant services. Section V discusses the need for and implementation of a KAS *broker*. Section VI is a brief conclusion and suggested way forward.

II. Review of SOA

A benefit of a service oriented architecture is that it is a backbone behind other architectural considerations whether the sensor system and knowledge extraction is performed locally or distributed. It is assumed in SOA that all services are interconnected in a mesh through some communications system which could be internal to a single computer, across multiple computers on a local area network (LAN), across a wide area network (WAN), or a cloud of cyber service providers.

SOA also supports the concept of *build a little, test a little* in that a set of core services can be specified and implemented initially along with other enhancements that can be added as supplementary services, e.g., toolboxes in Matlab for signal processing, libraries of conventional data fusion processes, etc. This also allows for a sufficient, but probably an inefficient initial implementation with service modules being replaced later with more efficient hardware or software as needed. An unrealized benefit of this approach is that once a threshold of services is reached, the entire system can be simulated to the degree of resolution available in the current implementation. This also means that services can later be moved from local inefficient computing resources to more efficient massively parallel, off-board cloud computing resources, or even direct hardware implementations (e.g., Field Programmable Gate Arrays (FPGA) or graphics processing unit (GPU)) of processes normally performed in software. That is, anything that is Turing computable can be equivalently implemented in hardware, software, or an integration of the two. The only tradeoff is between computational speed and the amount of hardware required to implement what is commonly thought of as a software process but is a mathematical process of symbol manipulation. An obvious example is the hardware or software implementation of a Fast Fourier Transform (FFT). An example of where an SOA has been applied to the integration of hardware (FPGA, GPU) and software functions for implementing deep learning functions is the Services-Oriented Deep Learning Architectures (SOLAR) by Wang et al. [6].

A hybrid mesh of SOA services also leads to an operational benefit in that it allows for graceful degradation of a knowledge acquisition system. As computing nodes or sensors become unavailable in real-time, other behaviorally equivalent services which are still available even if at a reduced accuracy, efficiency, or capacity, can be accessed to provide the requisite service.

Josuttis [7] states that a service oriented architecture in its most general sense is a loosely-coupled architecture which focuses on reducing system dependencies and is designed to meet the business or operational needs of an organization. Rimland and Llinas [8] state that "[t]he key tenet of SOA is that certain distributed applications are best implemented using a multitude of stateless, loosely coupled software components that each provides functionality in the form of a service." A comprehensive literature review of SOA in general is presented by Niknejad et al. [9] which includes thirteen definitions of SOA by various authors and concludes that "the most controversial factor among [SOA] researchers was SOA Governance [within an organization]." A second recent tutorial paper by Savaliya et al. [10], discusses the general vision and key enablers of SOA as well as a particular health care case study which integrates blockchain technology and Web 3.0 . General design principles are listed by Erl [11] as requiring: a *standardized service contract* that would provide the details about the functionalities offered by the service; *loose coupling* with minimal reliance on other services; *service abstraction* in that only the essential details about the service are available with the implementation being hidden; *service autonomy* with the logic of the service being entirely under the control of the service itself; *stateless implementation* meaning that the services should be transparent between the states; *service discoverability* where the services should become available quickly and effortlessly through the use of a service registry; *service composability* where it is easy to integrate the microservices to solve a significant problem; and finally, *service interoperability* insuring that each service is compatible with all the other services.

Different authors have characterized the properties of SOAs in a variety of manners. Papazoglou and Heuvel [12] characterized the properties of the services themselves as self-contained, platform independent, dynamically located and remotely invoked. They went on to characterize the architecture as service compositions which are divided into three categories, *fixed*, *semi-fixed*, and *explorative compositions*:

> Fixed service compositions require that their constituent services be synthesized in a fixed (pre- specified) manner. Semi-fixed compositions require that the entire service composition is specified statically but the actual service bindings are decided at run time. Finally, explorative compositions are generated on the fly on the basis of a request expressed by a client (application developer). [12]

In contrast to the distributed approach of [12], Benatallah and Nezhad [13] advocate a layered approach which is user-centric [14] while positing that there is still an integration and control problem. To this end, Web 2.0 [15] has been developed. It will be seen later that for KARPP we advocate the explorative composition approach wherein the user can provide services along with a response to a query as to how to use the service allowing for dynamic construction and alteration of the SOA architecture. Furthermore, we advocate a microservice architecture as defined in [16] which implements fine-grain services that act independently of one another such that failure of one service does not cascade into a cascade of failure across other services.

We continue here with specifics relevant to the application of SOA to knowledge acquisition in a government, or military context rather than a civilian enterprise such as a smart building [17], wireless body sensor network for a mobile medical intelligence platform [18], or manufacturing control environment [19].

While not advocating a services-wide standardization as in [20], Bassil [21] has decomposed SOA for weaponry and battle command and control into three elementary tiers: *client tier* referring to any computing equipment, a *server tier* that delivers the basic functionalities, and a *middleware tier* that promotes interconnectivity among the entities. Bassil also points out the six benefits of a decentralized core as: seamless integration, reusability, scalability, maintainability, survivability, and interoperability. Similar to other approaches, Bassil advocates web services and an Enterprise Service Bus (ESB) as well as the interfaces which would be required to implement such an approach, particularly among the various services.

In contrast, Hilal et al. [22], advocate a blackboard architecture for implementing a sensor management SOA based on three views: functional, physical, and interactional. They assume all sensor processing is done locally within the sensor resulting in a decision that is fused with other sensor decisions at a higher level called sensor resource management.

III. Extensible approach to designing KAS

Recognizing the dynamic nature of most knowledge acquisition systems, it seems appropriate to allow for a natural expansion and contraction of the mesh of services available to KARPP. To this end, we define a common unit of service, a knowledge aggregation service (KAS) which is further decomposed into its capabilities and access methods. As a KAS becomes available to an enterprise business system (EBS), it announces its services through a knowledge broker service (described later). This announcement includes its interface methods which comply with an industry standard as well as a reply to a particular query which describes the capabilities of the service much as is presented by Amoiridis et al. [23] although their application is focused on data acquisition for the compact muon solenoid particle detector. Their SOA is based on containers which encapsulate the various services as Docker [24] images thereby allowing them to be implemented on a variety of machines distributed through a WAN or LAN or within a single computer. We also believe that containerization of microservices is a desirable approach for KARPP.

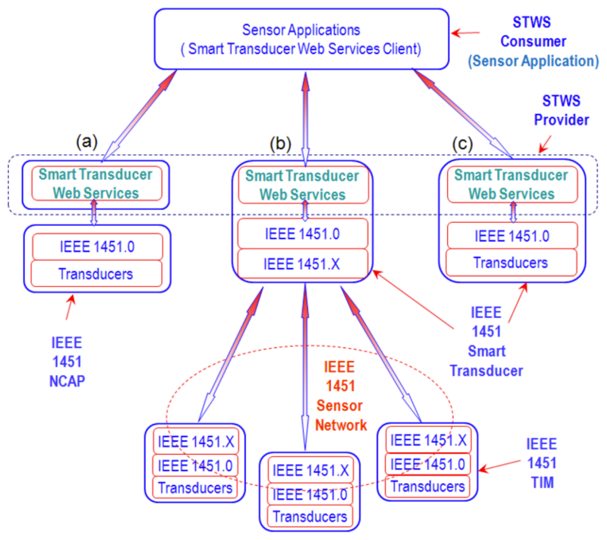

Figure 2 STWS- A unified Web service for IEEE 1451 smart transducers *[27]*.

The difficulty of including existing Representational State Transfer (RESTful) web services which are often used in publicly available APIs can be resolved by one of two methods. If the services are OpenAPI specification compliant, then the specifications are machine-understandable. If they are not, then the use of a publicly available software environment and the *parseText* function of Andročec and Tomašić [25] can produce a document body text [25]. An extensive review of the web service discovery literature is available in Tokmak et al. [26].

IV. Relevant KAS services

In order to implement KARPP utilizing microservices SOA, we will use the generic term knowledge aggregation services (KAS) as an overall term to describe the collection of individual services. The following is a more inclusive ontology of KAS services than those outlined in the KARPP paper [1].

A. *Ontology of KAS*

Services available in KARPP are KAS implementations and support implementations of the four services of which it is comprised: acquisition, representation, processing, and presentation plus support services. Each of these can be further defined in terms of the services that are available in each of these KAS but these details are beyond the scope of this paper and are a work in progress.

1) AcqS: sensor/data Acquisition Service

Acquisition services (*AcqS*) can be further subdivided into services that acquire data from sensors of (physical) observable phenomena such as infrared (IR), radar, or electronic support measures (ESM), and knowledgebase services such as geospatial maps, translators, or social network relationships. The former can acquire kinematic data leading to state estimates and the latter may contain context for interpretation of the estimates.

A simple and straightforward example of SOA and an *AcqS* as applied to IEEE 1451 smart transducers is shown in Figure *2* [27]. While this is a specific hardware sensor example, it shows the hierarchy of services ranging from the sensor through the standard 1451 interface, to the web services and finally to the consumer's use of sensor applications and web services client.

Acquisition services (*AcqS*)

1. Physical kinematic sensor services such as:
   - Radars and Lidars (call: search, track, ID, pointing angle; return: data for kinematic state estimation by another processor service)
   - Non-ionizing electromagnetic (EM) imaging sensors (call: search, track, ID, pointing angle; return: neighborhoods of pixels for identification by other processor service)
   - Ionizing EM radiation sensors (call: radiation type, location; return: spatial location of source, strength for analysis by another processor service)
   - Acoustic sensors (call: acoustic band, pointing angle; return: detection data for analysis by processor service)
2. Cyber sensor services

- Social network and directed entity graph services (call: entity; return associates, strength of association, direction)
- LAN, WAN sensors (call: source; return: connected nodes, bandwidth; return: connected nodes for analysis by another process service)

3. Database sensor services

- Generic database search (call: database, SQL query; return: query results, quality)
- Target trackfile (call: target ID or location; return: target kinematic state, existence of target, quality of track)

2) RepS: Representation Service

There are many ways of representing knowledge both for the internal processing of the data as well as use later by the presentation service for transferring the knowledge to a decision maker [28]. Let's assume that the users of the presentation layer of a sensing system are comprised of only two different types of individuals: the *analysts* and the *decision makers*. We can differentiate between them by the scope of their intrinsic, mental knowledge. The analyst is far more knowledgeable about the sensing system, its capabilities, and the underlying processes that create the displayable knowledge and even the details of a situation. The analyst may also interact with the sensing system either to control it manually or to change the relative values of the best next collection opportunity (BNCO) valuation method. To put it more succinctly, the analyst is interested in the factual situation knowledge rather than the situation awareness or the meaning of that situation knowledge. On the other hand, the decision maker is less interested in the situation knowledge and more interested in the implications of the acquired knowledge with respect to his intrinsic, mental knowledge of his world.

a) Representation for the Analyst User KAS

- Goal lattice (GL) valuation of knowledge requests (call: relative mission value data, knowledge to evaluate; return: relative values of evaluated knowledge)
- Display of relative value (call: relative value function; return: GUI display of relative values)
- Relative value modification service (call: current knowledge values; return: GUI or text input for modifying relative values)
- Bayesian network (BN) of situation knowledge (call: network nodes, probabilities, and conditional probabilities; return: Bayesian network)
- Credal network (CN) of situation knowledge (call: network nodes, possibilities; return: credal network)
- Geospatial (call: targets coordinates, coordinate system; return: geospatial representation, coordinate system)
- Directed graph of social network (call: nodes, directions; return: directed graph)

b) Representation for the Decision Maker User KAS

- Situation state predictions (call: situation BN or DS, possible actions, costs; return: mission valued predicted state, uncertainty in the situation of interest)
- Spider diagram of knowledge qualities (call: current qualities, desired qualities; return: donut chart)
- Adversarial goal lattice (call: current goal lattice estimate, observed actions of adversaries; return: estimated GL

3) ProcS

We assume further that there may be multiple methods, algorithms, or services which can be used to process the several sources of knowledge in order to aggregate them into a larger scoped knowledge. The choice of which process service to apply to the several sources of knowledge is a significant part of the knowledge acquisition control mechanism. That is, resource management within each KAS is meant to apply to both the selection of sources and their modes of acquiring knowledge as well as the subsequent processing of this knowledge.

The effectiveness and timeliness of knowledge processing is intimately controlled by the amount and method of computation which is available to the KAS. A deployed unmanned autonomous vehicle (UAV) may be severely constrained not only computationally in its payload, but also in the amount of fuel available to generate computational energy as well as propulsion. Even lighter helicopter like drones are severely limited in battery power affecting their useful flight time leading one to prefer to downlink all data for off-board processing as a service.

a) Processors for the Analyst User KAS

At the KAS levels between the knowledge source sensors and the knowledge requestors is the local control of the knowledge sources, the resource management. Resources are both the sources of the knowledge and the knowledge processing capabilities. The two need to be matched to extract the maximum requested knowledge with the desired uncertainty and value as measured by the non-commensurate requested qualities that are passed down to the KAS from the next higher level.

To this end, there are two services that need to be implemented. They are dynamic and populated by the knowledge source service and the knowledge processing service.

- Available knowledge source service (call: desired knowledge type; return: knowledge sources available that can provide that knowledge).
- Available knowledge processing table (call: desired knowledge processing type; return: knowledge processes available that can process the knowledge).
- Best next collection opportunity (BNCO) evaluator (call: uncertainties, probabilities, mission value, timeliness requirement; return: BNCO)
- Kinematic state estimator (call: sensor data from acquisition KAS; return: kinematic state estimate with error covariance matrix), e.g., Kalman filter or alpha-beta filter
- Dynamic model state estimators (call: sensor data from acquisition KAS, set of models; return: kinematic state estimate with error covariance matrix and model used), e.g., multiple hypothesis tracking (MHT) [29], interacting multiple model (IMM) [30], variable structure IMM (VS-IMM) [31]
- Image processor (call: imaging sensor data, desired function; return: target location, ID, uncertainty)

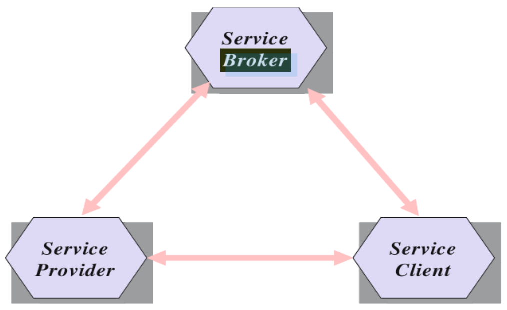

Figure 3 Role of a service broker [12].

- Natural language processor (NLP) (call: unstructured text; return: parsed text in standardized, machine readable form)

b) Processors for the Decision Maker User KAS

- Game theoretic state predictor (call: own state, adversarial state, mission values, context; return: possible courses of action, possible outcomes of selected actions)
- Goal lattice relative value predictor (call: mission values, context, chosen course of action; return: revised GL values)

4) PresS: Presentation Service

Impedance matching is a well-known method for maximizing the transfer of power from one electrical circuit (the source) to a second electrical circuit (the load). Analogous to this in a knowledge management system is a knowledge impedance matching from a source of data such as a sensor to a load such as a data fusion algorithm, or from the output of a fusion algorithm to a knowledge display interface, or from a display to a user. In the case of different users, different knowledge impedance matching methods are needed which are matched to the users' intrinsic knowledge.

Let's again assume that the users of the presentation layer of a sensing system are comprised of only two different types of individuals, the *analysts* and the *decision makers*. We can differentiate between them by the scope of their intrinsic, mental knowledge. The analyst is far more knowledgeable about the sensing system, its capabilities, and the underlying processes that create the displayable knowledge and even the details of a situation.

a) Presentation services for the analyst

- Analytical tools available to the analyst at the presentation level (call: input data, analysis to perform, format of returned data; return: graphs, texts, images)
- Knowledge trend (call: knowledge history; return: knowledge temporal statistics in graph or tabular form)
- GL graphical display with GUI interaction (call: mission context sensitive GL; return: interactive graphical GL display for real measurable action values)
- KARPP performance display(call: none; return: graphical display of performance relative to desired mission GL)

b) Presentation services for Decision Maker

- Variable resolution geospatial current situation display (call: current BN or CN situation estimate; return: graphical abstraction of situation)
- Variable resolution geospatial prediction situation display (call: proposed course of action; return: graphical abstraction of predicted situation, uncertainty in prediction)
- Combat statistics (call: battle damage assessment (BDA); return: tabular and graphical statistics of own and adversarial forces)
- Spider diagram display of uncertainty in the situation of interest (call: qualities from situation BN or CN; return: Donut display of mission qualities)
- GL graphical display with GUI interaction (call: mission context sensitive GL; return: interactive graphical GL display for topmost mission goals)

5) Support services

Knowledge presentation can also be viewed as a pre-user interface issue. That is, the requestor of the knowledge may need it to be returned in a particular form. This leads to the question of who does the formatting, the requestor or the supplier? At this stage of development of KARPP services and KAS, it is not clear, although a reasonable approach would be to define in the interface between two KAS the format in which it is desired to receive the knowledge. It would appear that the supplier of the knowledge would be in a better position to determine how to transform its acquired knowledge into a form that is requested. Also this approach provides for spiral development in which there is an enumerated list of knowledge formats, with the exact elements of this list to be expanded as capabilities of the KAS service are developed and expanded to meet unforeseen needs and capabilities.

- Knowledge format conversion services (call: input knowledge, desired format; return: reformatted knowledge
- Geophysical map services (call: geophysical map source, desired data; return: extracted geophysical data)
- Soft/Hard data fusion services (call: hard data, soft data, context; return: fused data)

V. KAS Broker

To this point we have assumed that there are knowledge aggregation services of various types that are available either locally or distributed without specifying how they are composited into an effective, dynamic mesh to achieve a knowledge acquisition end. We view KARPP as not a single architecture in itself, but a paradigm for autonomously constructing context and platform sensitive systems in terms of their connectivity of services and level of use from the highest levels of decision making to the lowest levels of data acquisition by observing a physical phenomenon or accessing a knowledge base. In essence, KARPP is a mesh of services whose connectivity is automatically particularized to a specific problem or context and which can dynamically change as a knowledge need evolves.

We envision this process to be a self-organizing connectivity (not specifically layered, but rather a mesh) whose connectivity is determined by the feedforward request for knowledge, the qualified knowledge request (QKReq), and the returned aggregated knowledge response (AKRes) which is provided by a KAS. Qualified in this context is meant to indicate the level of *quality* of the requested knowledge. This also assumes a dynamic communications backbone which enables the maximum connectivity among the mesh of services in both a

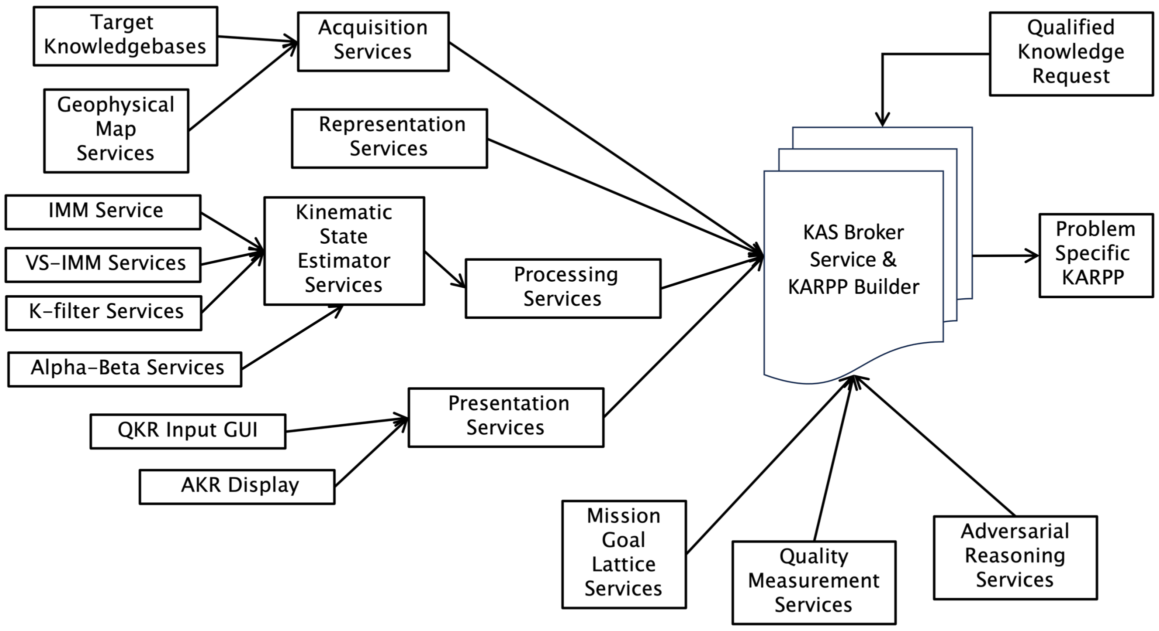

Figure 4 Knowledge Aggregation Service broker which responds to requests for available services with access links and formal interface requirements.

tactical and strategic configuration as well as dynamic reconfiguration [32].

This dynamic knowledge services broker (KSB) (an abbreviated concept is shown in Figure *4*) serves multiple purposes including managing which KAS have access to which level of classified data, organizing services by function, and dissemination of services capabilities and I/O requirements. To enable this functionality, it is necessary that each KAS provide and broadcast its capabilities in a standardized format such as the sensor modeling language [33] [34] utilized by the open geospatial consortium.

This dynamic interconnectivity is enabled by each KAS having access to a KAS broker which has been populated in one of two ways. A service can analyze and characterize currently available services (discussed earlier) or populated by announcements transmitted by newly available services to the broker. The term *broker* which we have expanded to KAS Broker was first introduced by [12]. A simple block diagram of the relationship between service providers/clients and the service broker is shown in Figure 3.

Qualified knowledge requests recognize the fact that a decision maker not only needs knowledge with which to make a decision but also needs that knowledge to meet some particular level of quality. Aggregated knowledge responses recognize the fact that the responses from any KAS will be particularized to the specific QKReq which is presented to it. Simultaneous QKReq from various KAS can result in knowledge resource

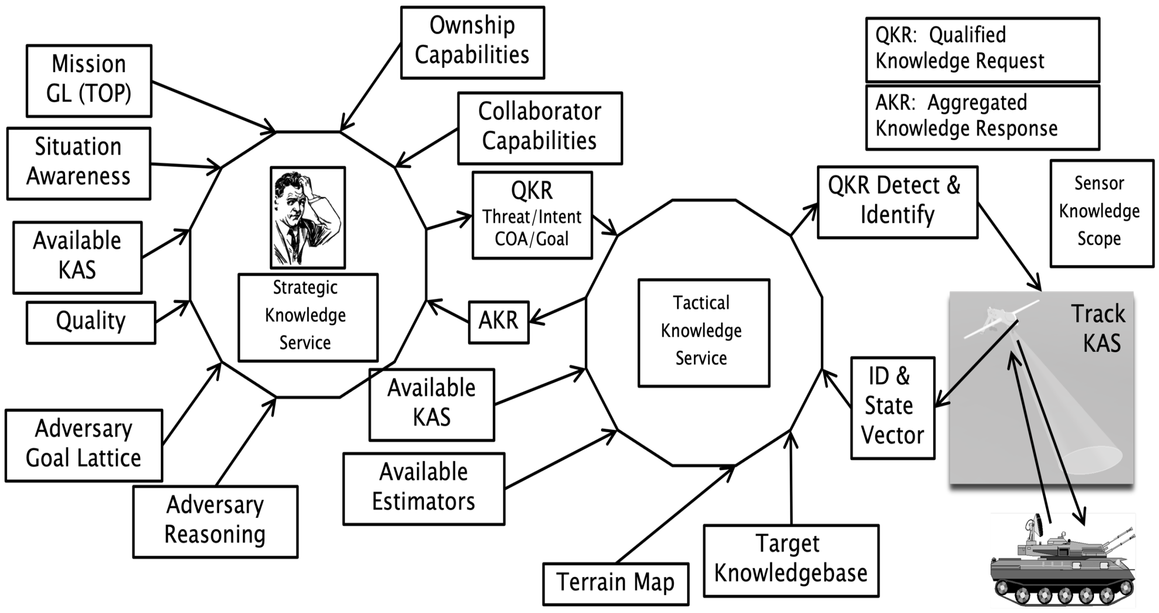

Figure 5 KARPP implemented as a simplified two layered KAS showing the relationship between a decision maker, a sensor, and the services needed to aggregate the knowledge.

conflicts at the lowest level but this has been shown to be effectively handled by the *information based sensor management* (IBSM) methodology which orders the knowledge needs based on maximizing the sensor *expected information value rate* (EIVR) [35].

Artificial intelligence (AI) has been used to automatically find services in order to build an SOA and Rodriguez et al. [36] review 69 of them. They also write that "[t]he main advantage of the use of AI stems from its efficient performance in dynamic, distributed, non-deterministic and uncertain environments; reducing the search space." as well as "[i]n general, a system for discovering services needs a service description language, a service selection means (i.e., matchmakers) and discovery architecture (e.g., decentralized [Peer to Peer (P2P)])." Additional web services discovery methods are described by Tokmak et al. [37].

Recognizing the problem of there being no single system architecture which will fit every situation assessment/awareness problem, and with reference to Figure *5*, we see that the decision maker is interfacing with a Strategic Knowledge KAS Service (SKS) which is dynamically connected to those other services which it needs to function. Part of this SKS is the knowledge of those other KAS and services to which it has access. All KAS need to be able to determine those services to which they have access else the architecture is static. Additionally, a KAS needs to be able to *come on-line* with no knowledge of its connectivity to various services. Initialization of each KAS is accomplished by its contacting a knowledge services broker (KSB) and requesting links to those services which it needs to aggregate knowledge.

VI. Conclusion and way forward

A review of service oriented literature has confirmed that it is an appropriate and useful architecture for implementing the dynamic, context sensitive KARPP architecture for situation management. Not only is it capable of implementing the envisioned mesh of knowledge aggregation services of the various types, but there also exist tools to automatically search and classify the various existing services that can be used as components of KARPP microservices.

Since KAR PP is an *architecture of architectures* it recognizes the fact that there is no single architecture which can meet all knowledge requirements and that the decision maker's environment and the sensor capabilities change in response to an adversary's demonstrated and observable behavior as well as the physical environment.

Implicit in the micro SOA implementation of KARPP is the interservice communications which is comprised of qualified knowledge requests (QKReq) and the returned knowledge in the form of an aggregated knowledge response (AKRes). These communications take various forms depending on the scope of the knowledge available to the KAS.

Future work will be directed towards the implementation of a skeleton SOA of microservices which will require the adoption of a standards based interface and implementation of basis KAS.

Acknowledgment

Research presented in this paper was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20220799DI.

References

[1] K. J. Hintz, J. Llinas and K. Sentz, "Knowledge Acquisition, Representation, Processing & Presentation (KARPP)," in *IEEE CogSIMA*, Montreal, Canada, 2024.

[2] Y. Zhao, P. Wen, L. Bai, X. Liu, Z. Wang and N. Wu, "Service-oriented Intelligent OODA Loop," in *26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Taiyuan, Taiwan, 2023.

[3] F. Boumahdi, H. Oqaibi, R. Chalal, H. Hentabli and A. Madani, "MDA4SOA: A new model driven architecture to supporting decision making in SOA," *Journal of King Saud University – Computer and Information Sciences,* vol. 35, no. 101544, 2023.

[4] V. Casola and A. Mazzeo, "SeNsIM-Web: A service based architecture for sensor networks integration," in *35th Annual Conference of IEEE Industrial Electronics (IECON 2009)*, Porto, Portugal, 2009.

[5] J. Tang, H. Ren, C. Yang, L. Shen and J. Jiang, "A WEBGIS FOR SHARING AND INTEGRATION OF MULTI-SOURCE HETEROGENEOUS SPATIAL DATA," in *2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Vancouver, BC, 2011.

[6] C. Wang, L. Gong, X. Li, Q. Yu, A. Wang, P. Hung and X. Zhou, "SOLAR: Services-Oriented Deep Learning Architectures-Deep Learning as a Service," *IEEE TRANSACTIONS ON SERVICES COMPUTING,* vol. 14, no. 1, pp. 262-273, 2021.

[7] N. M. Josuttis, SOA in practice: the art of distributed system design, O'Reilly Media, Inc., 2007.

[8] J. C. Rimland and J. Llinas, "Network and Infrastructure Considerations for Hard and Soft Information Fusion Processes," in *15th International Conference on Information fusion, Fusion2012*, Singapore, 2012.

[9] N. Niknejad, W. Ismail, I. Ghani, B. Nazari, M. Bahari and A. R. B. C. Hussin, "Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation," *Information Systems,* vol. 91, no. July, 2020.

[10] D. Savaliya, S. Sanghavi, V. K. Prasad, P. Bhattacharya and S. Tanwar, "Tutorial on Service-Oriented Architecture: Vision, Key Enablers, and Case Study," in *Proceedings of International Conference on Recent Innovations in Computing (ICRIC)*, Jammu, India, 2022.

[11] T. Erl, SOA: principles of service design, Upper Saddle River, NJ: Prentice-Hall, 2008.

[12] M. P. Papazoglou and W.-J. van den heuvel, "Service oriented architectures: approaches, technologies and research issues," *VLDB Journal ,* no. 16, pp. 389-415, 2007.

[13] B. Benatallah and H. R. M. Nezhad, "Service Oriented Architecture: Overview and Directions," in *Advances in Software Engineering*, Berlin, Springer, 2008, pp. pp 116-130.

[14] IBM, Microsoft, "Web Services Framework," W3C Workshop, 11-12 April 2001. [Online]. Available: https://www.w3.org/2001/03/WSWS-popa/paper51. [Accessed 21 January 2024].

[15] T. O'Reilly, "What is Web 2.0," O'Reilly, 30 September 2005. [Online]. Available: https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html. [Accessed 21 January 2024].

[16] M. L. Fanomezana, A. M. Rapatsalahy, N. . R. Razafindrakoto and C. Bădică, "Proposed Methodology for Designing a Microservice Architecture," in *2022 23rd International Carpathian Control Conference (ICCC)* , Virtual, 2022.

[17] L. Chamari, E. Petrova and P. Pauwels, "An End-to-End Implementation of a Service-Oriented Architecture for Data-Driven Smart Buildings," IEEE Access, 2023.

[18] A. Petrenko, "Approaches for WSN (Wireless Sensor Networks) Standardization and their Interoperability in Combining into a Global Network," in *2023 IEEE East-West Design & Test Symposium (EWDTS)* , Batumi, Georgia, 2023.

[19] K. Xu, T. Wang and L. Cheng, "Service Recommendation of Industrial Software Components Based on Explicit and Implicit Higher-Order Feature Interactions and Attentional Factorization Machines," *Applied Sciences,* vol. 13, no. 10746, 2023.

[20] M. M. Jamjoom, A. S. Alghamdi and I. Ahmed, "Service Oriented Architecture Support in Various Architecture Frameworks: A Brief Review," in *World Congress on Engineering and Computer Science*, San Francisco, CA, 2012.

[21] Y. Bassil, "Service-Oriented Architecture for Weaponry and Battle Command and Control Systems in Warfighting," *International Journal of Information and Communication Technology Research,* vol. 2, no. 2, 2012.

[22] A. R. Hilal, A. Khamis and O. Basir, "HASM: A Hybrid Architecture for Sensor Management in a Distributed Surveillance Context," in *2011 International Conference on Networking, Sensing and Control*, Delft, Netherlands, 2011.

[23] V. Amoiridis and 20 more, "Towards a container-based architecture for CMS data acquisition," in *CHEP2023 26th International Conference on Computing in High Energy Physics and Nuclear Physics*, Geneva, Switzerland, 2023.

[24] Docker.com, "Docker Resources," 2023. [Online]. Available: https://www.docker.com/resources/what-container/. [Accessed 27 January 2024].

[25] D. Andročec and M. Tomašić, "Using GPT-3 to Automatically Create RESTful Service Descriptions," in *4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Plovdiv, Bulgaria, 2023.

[26] A. V. Tokmak, A. Akbulut and C. Catal, "Web service discovery: Rationale, challenges, and solution directions," *Computer Standards & Interfaces ,* vol. 88, no. 103794, 2024.

[27] E. Y. Song and K. B. Lee, "Service-oriented Sensor Data Interoperability for IEEE 1451 Smart Transducers," in *International Instrumentation and Measurement Technology Conference (I2MTC)*, Singapore, 2009.

[28] D. Norman, Design for a Better World, Ch. 11, Cambridge, MA: MIT Press, 2023.

[29] T. Kirubarajan, Y. Bar-Shalom, W. D. Blair and G. A. Watson, "IMMPDA solution to benchmark for radar resource allocation and tracking in the presence of ECM," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 34, no. 3, pp. 1023-1036, 1998.

[30] Y. Bar-Shalom and X. R. Li, Estimation and Tracking: Principles, Techniques and Software, Storrs, CT: Artech House, 1993.

[31] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati and I. Kadar, "Ground Target Tracking with Variable Structure IMM Estimator," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS,* vol. 36, no. 1, pp. 26-46, 2000.

[32] A. Aloisio, M. Autili, A. D'Angelo, A. Viidanoja, J. Leguay, T. Ginzler, T. Lampe, L. Spagnolo, S. Wolthusen, A. Flizikowski and J. Sliwa, "TACTICS: TACTICal Service Oriented Architecture," in *3rd International Conference in Software Engineering for Defence Applications*, Rome, Italy, 2014.

[33] Open Geospatial Consortium, "OGC® SWE Implementation Maturity Engineering Report," 11 September 2013. [Online]. Available: www.opengisl.net/doc//er/swe-maturity. [Accessed 2 February 2018].

[34] Open Geospatial Consortium, "OGC® SensorML: Model and XML Encoding Standard," 2014. [Online]. Available: http://www.opengeospatial.org/standards/sensorml. [Accessed 20 02 2018].

[35] K. Hintz, Sensor Management in ISR, Boston: Artech House, 2020.

[36] G. Rodriguez, A. Soria and M. Campo, "Artificial intelligence in service-oriented software design," *Engineering Applications of Artificial Intelligence,* vol. 53, pp. 86-104, 2016.

[37] A. V. Tokmak, A. Akbulut and C. Catal, "Web service discovery: Rationale, challenges, and solution directions," *Computer Standards & Interfaces,* vol. 88, no. 103794, 2024.