

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC.,

Petitioner

IPR2025-00624

U.S. Patent No. 8,040,232

**DECLARATION OF HENRY HOUH,
UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION
FOR *INTER PARTES* REVIEW**

TABLE OF CONTENTS

I.	INTRODUCTION.....	4
II.	QUALIFICATIONS AND PROFESSIONAL EXPERIENCE.....	6
III.	LEVEL OF ORDINARY SKILL IN THE ART	12
IV.	RELEVANT LEGAL STANDARDS	13
V.	TECHNICAL BACKGROUND	14
	A. Ubiquitous Sensor Networks.....	14
	B. Network Gateways	15
	C. Communication Protocols and Application Program Interfaces (APIs)	16
VI.	THE '232 PATENT.....	17
VII.	CLAIM CONSTRUCTION.....	20
VIII.	IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE...	21
	A. Ground 1: Claims 1-12 are obvious under §103 over Jakobson in view of Tsetsos.....	21
	1. Summary of Jakobson	21
	2. Summary of Tsetsos	25
	3. Reasons to Combine Jakobson and Tsetsos	26
	4. Detailed Analysis of the Claims.....	29
	a) Claim 1	29
	b) Claim 2	77
	c) Claim 3	88
	d) Claim 4.....	104
	e) Claim 5	113
	f) Claim 6.....	127
	g) Claim 7	128
	h) Claim 8.....	129

i)	Claim 9	130
j)	Claim 10	130
k)	Claim 11	132
l)	Claim 12	133
IX.	CONCLUSION	136

I, Henry H. Houh, do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Amazon.com, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 8,040,232 (“the ’232 patent”) to Oh et al.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1-12 (“the Challenged Claims”) of the ’232 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- a. the ’232 patent, Ex.1001;
- b. the prosecution history of the ’232 patent (“’232 File History”), Ex.1002;
- c. U.S. Patent Application Publication No. 2005/0222811 to Jakobson et al. (“Jakobson”), Ex.1005;

d. Commercial Wireless Sensor Networks: Technical and Business Issues, by Tsetsos et al. (“Tsetsos”), Ex.1006;

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards, including the standard for obviousness; and my own knowledge and experience based upon my work in the fields of digital networking and data processing as described below, as well as portions of the following additional materials:

a. Newton’s Telecom Dictionary, 20th ed. (2004) (“Newton”), Ex.1007;

b. Computer Networks, 4th ed. (2004), Andrew S. Tanenbaum (“Tanenbaum”), Ex.1008;

c. Tomioka Katsumi, “Ubiquitous Sensor Network System,” C&C User Forum & iEXPO2005 (“Katsumi”), Ex.1016;

d. Daniele Puccinelli et al., “Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing,” IEEE Circuits and Systems Magazine, (3rd Quarter, 2005) (“Puccinelli”), Ex.1017;

e. Irfan A. Essa et al., “Ubiquitous Sensing for Smart and Aware Environments,” IEEE Personal Communications, (October 2000) (“Essa”), Ex.1018;

f. Peter Schramm et al., “A Service Gateway for Networked Sensor Systems,” IEEE Pervasive Computer (1st Quarter, 2004) (“Schramm”), Ex.1019;

g. Gaurav Gupta et al., “Fault-Tolerant Clustering of Wireless Sensor Networks,” IEEE Wireless Communications and Networking (May 2003) (“Gupta”), Ex.1020;

6. Unless otherwise noted, all **emphasis** in any quoted material has been added. Claim terms are *italicized*.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Ex.1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (“MIT”) in 1998. Beforehand, I received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science degree in Physics in 1990, all from MIT.

9. I am currently self-employed as an independent technical consultant. Until October 2022, I was also president of a company that provides supplemental

science, technology, engineering, and mathematics (“STEM”) education to children of all ages.

10. I first entered telecommunications in 1987 when I worked as a summer intern at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. I continued to work at AT&T Bell Laboratories as part of this MIT program. While at MIT, I was a teaching assistant (“TA”) in the Electrical Engineering and Computer Science Department’s core Computer Architectures course. I first was a TA as a senior for a role typically reserved for graduate students. I later became head TA. The course covered various topics in computer architectures. As a TA, I helped write homework assignments, lab assignments, and exams. I also taught in the recitation sections.

11. Later, as part of my doctoral research at MIT from 1991-1998, I was a research assistant in the Telemedia Network Systems (“TNS”) group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit network and created applications that ran over the network. Example applications included ones for remote video capture, processing, and display of video on computer terminals. In addition to working on the design of core network components, designing and building the high-speed links, and designing and writing the device drivers for the interface cards, I also set up the group’s web server.

12. I also helped to build the web pages that initiated the above-mentioned video sessions via a web interface. Vice President Al Gore visited our group in 1996 and received a demonstration of—and remotely drove—a radio-controlled toy car with a wireless video camera mounted on it that was built by our group. This toy car device received commands transmitted over a network from a remote computer, and video data from the toy car was transmitted wirelessly then over a computer network back to the user controller. On occasion, we allowed users visiting our web site to drive the toy car from their remote computer while they watched the video on their computer. The video stream was encoded by TNS-designed hardware, streamed over the TNS-designed network, and displayed using TNS-designed software.

13. I defended and submitted my Ph.D. thesis, titled “Designing Networks for Tomorrow’s Traffic,” in January 1998. As part of my thesis research, I analyzed local area and wide area flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time.

14. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines. NBX was later acquired by 3Com Corporation and the phone system is still used today by numerous businesses.

15. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system. The protocol was used for all signaling in the phone system, including for the setup of conference calls.

16. The NBX system also featured a computer interface for initiating phone calls, which could also initiate conference calls. The NBX system also supported the Telephony Application Programming Interface (“TAPI”) that allowed other computer programs to integrate with our system telephony features. We obtained U.S. Patent No. 6,967,963, entitled “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” as part of this work.

17. From 1999-2004, I was employed by Empirix or its predecessor company, Teradyne. Empirix was a leader in test tools for telecommunications protocols and systems, providing functional testing tools as well as load testing tools. From 2000-2001, I conceived and built a test platform for testing Voice-over-IP (VoIP). The first application on this new test platform was a cloud emulator for simulating the effects of transmitting VoIP over a busy network.

18. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN’s speech

recognition algorithms to search through the spoken words in audio and video segments. While I was Vice President of Operations and Technology, PodZinger followed its initial prototype with a full streaming audio and video search solution. I also created a social networking web site, which BBN sold to a venture-funded startup company. In the process of creating the web site, I designed and specified the authentication and authorization protocols.

19. I have been awarded several United States patents, and I have several patent applications pending including the following examples:

- U.S. Patent No. 7,975,296, “Automated security threat testing of web pages”;
- U.S. Patent No. 7,877,736, “Computer language interpretation and optimization for server testing”;
- U.S. Patent No. 7,801,910, “Method and apparatus for timed tagging of media content”;
- U.S. Patent 7,590,542, “Method of generating test scripts using a voice-capable markup language”;
- U.S. Patent No. 6,967,963, “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data”;

- U.S. Patent Application Publication No. 20070106685, “Method and apparatus for updating speech recognition databases and reindexing audio and video content using the same”;
- U.S. Patent Application Publication No. 20070106693, “Methods and apparatus for providing virtual media channels based on media search”;
- U.S. Patent Application Publication No. 20070106760, “Methods and apparatus for dynamic presentation of advertising, factual, and informational content using enhanced metadata in search-driven media applications”;
- U.S. Patent Application Publication No. 20070112837, “Method and apparatus for timed tagging of media content”;
- U.S. Patent Application Publication No. 20070118873, “Methods and apparatus for merging media content”; and

20. U.S. Patent Application Publication No. 20090222442, “User-directed navigation of multimedia search results.”

21. I have reviewed the '232 patent, and relevant excerpts of the prosecution history of the '232 patent. Based on my experience and education, I believe that I am qualified to offer opinions as to the knowledge and level of skill of

one of ordinary skill in the art at the time of the invention of the '232 patent (which I further describe below).

III. LEVEL OF ORDINARY SKILL IN THE ART

22. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

23. I have been informed that the '232 patent claims priority on its face to Korean Patent Application No. 10-2006-0084353, which was filed September 1, 2006.

24. A POSITA in the field of the '232 patent, on or around September 2006, would have been someone knowledgeable and familiar with digital networking and data processing arts that are pertinent to the '232 patent. That person would have a bachelor's degree in computer engineering, computer science, or equivalent training, and approximately two years of experience working in the fields of digital networks and data processing software. Lack of work experience can be remedied by additional education, and vice versa.

25. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my experience and

what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the alleged priority date of the '232 patent (i.e., September 1, 2006). Unless otherwise stated, when I provide my understanding and analysis below, it is consistent with the level of a POSITA prior to the alleged priority date of the '232 patent.

IV. RELEVANT LEGAL STANDARDS

26. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '232 patent, I am relying on certain basic legal principles that Amazon's counsel has explained to me. These principles are discussed below.

27. I understand that prior art to the '232 patent includes patents and printed publications in the relevant art that predate the priority date of the '232 patent.

28. I have been informed by Amazon's counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by Amazon's counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

29. I have been further informed by Amazon’s counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness of claimed subject matter. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

V. TECHNICAL BACKGROUND

A. Ubiquitous Sensor Networks

30. “Ubiquitous Sensor Networks (USNs)” refers to a paradigm in which sensor nodes are integrated into a variety of environments to monitor, collect, and transmit real-time data. Katsumi, Abstract, 1; Puccinelli, Abstract, 2. Rather than describing a specific type of sensor network or communication protocol, a USN is an implementation or application of sensor networks that can accommodate a spectrum of network scales, architectural designs, and communication protocols.

Katsumi, Abstract, 1; Essa, 1 (“Ubiquitous Sensing”); Tsetsos, 3, 5 (describing heterogeneous wireless sensor networks for ubiquitous services), 6 (“ubiquitous context-aware services”). In this way, USNs can be tailored for specific applications. Katsumi, Abstract.

31. The dynamic and heterogeneous nature of USNs means that sensor nodes may vary widely in terms of their hardware capabilities, power sources, and operational lifespans. USNs may operate in unpredictable or resource-limited environments, or in resource-dense environments. Accordingly, “ubiquitous” refers to the pervasiveness of these networks and their ability to evolve and integrate into a myriad of scenarios. Katsumi, Abstract; Tsetsos, 3, 5, 6.

B. Network Gateways

32. Gateway devices serve as intermediary devices in network architectures, bridging the gap between diverse, localized data sources and larger (including global) networks such as the Internet. Schramm, 5; Tsetsos, 3; Gupta, 1-2; Tanenbaum, 25. These devices collect, process, and aggregate data from a myriad of sources—ranging from simple environmental sensors and industrial monitoring units to more complex endpoints—and then transmit this consolidated information to servers or other platforms that process the data to provide a service. Schramm, 2; Tsetsos, 4. Gateway devices may include relatively greater

communication and data processing capabilities than the networked devices to which they are connected (e.g., sensors). Tsetsos, 3; Schramm, 2-3; Gupta, 2.

33. Gateway devices handle data streams that originate from devices using different data formats and communication standards, including proprietary ones. Schramm, 2. By executing these tasks at the network's edge, gateways can reduce data processing and translation capabilities at other data processing systems upstream, and can facilitate scalability by allowing increasingly large numbers of devices using different communication protocols and formats to connect to a larger network and provide data to the same data processing systems. Schramm, 2-3.

C. Communication Protocols and Application Program Interfaces (APIs)

34. Digital communication protocols, such as TCP/IP, define the rules and standards governing how data is packaged, transmitted, received, and interpreted between devices, ensuring that information is exchanged reliably regardless of hardware or geographic location. The Internet Protocol (IP) handles addressing and routing, with associated routing protocols determining paths for data packets to traverse a complex network of interconnected nodes. The Transmission Control Protocol (TCP) works at a higher layer to ensure that data are delivered accurately and in order, establishing error-checking and retransmission mechanisms that guarantee data integrity and reliability. These protocols form a robust and scalable

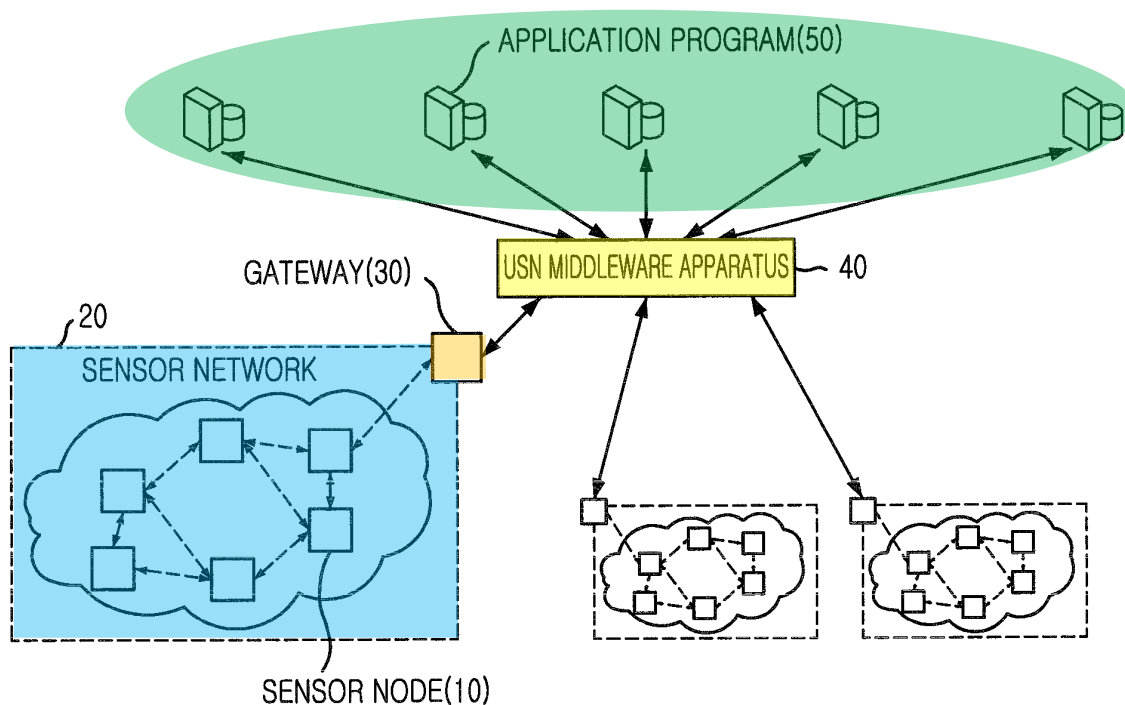
backbone for a wide range of communication networks that exchange digital information. Tanenbaum, 41-44; 532-539.

35. Building upon these networking communication protocols are Application Programming Interfaces (APIs), which serve as higher-level abstractions that simplify and standardize interactions between different components in a network. Tsetsos, 3. While TCP/IP provides the low-level mechanisms for moving data between devices, APIs define a set of methods and data structures that application developers can use to access the functionality of a service, without needing a detailed knowledge of the intricacies and inner workings of that service. Tsetsos, 3. The API specifies what inputs are required and what outputs will be provided. The API provides a set of pre-defined functional messages for software developers to interact with the service. This facilitates the development of new services that work across diverse systems and service providers, and helps maintain consistency and security across applications.

VI. THE '232 PATENT

36. The '232 patent describes a “Ubiquitous Sensor Network (USN) middleware apparatus” that receives sensor data from “heterogeneous sensor networks” and provides “a user with information related to [sic] environment where the sensor network observes.” ‘232 Patent, Abstract. FIG. 1 shows an example of the USN middleware apparatus (yellow) receiving data from a sensor

network (blue) via a gateway device (orange) and providing an information service to “application programs” (green).

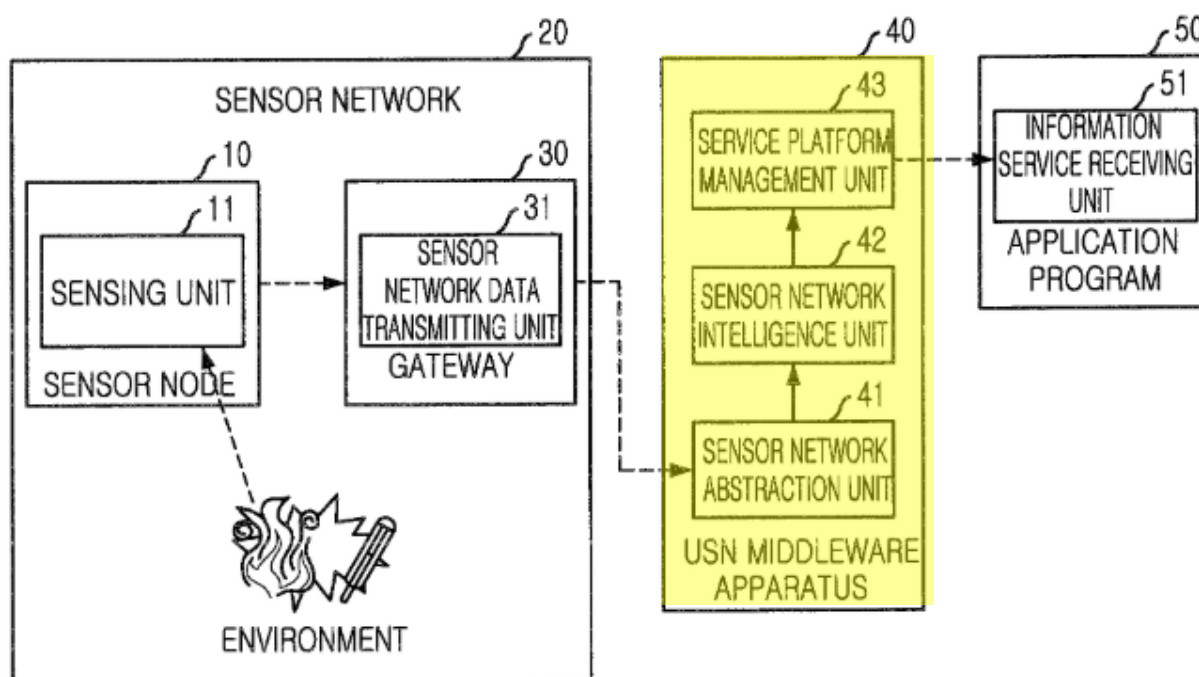


'232 Patent, FIG. 1 (annotated)

37. The sensor nodes in the sensor network provide sensor data representing environmental conditions like “temperature, humidity, heat, and infrared rays.” ‘232 Patent, 4:50-51. A gateway device collects the data from the various sensors and transmits the data to the USN middleware as a message using a communication protocol, such as the “Transmission Control Protocol/Internet Protocol (TCP/IP).” ‘232 Patent, 4:35-37.

38. The USN middleware apparatus, highlighted in yellow in FIG. 2 below, includes submodules or units that extract the sensor data and perform a

sequence of data analysis operations that include “extract[] low-level information, i.e., object properties and environment state information,” and generate “high-level information such as context aware information and conditional events about a wide area.” ‘232 Patent, 4:20-25, 3:16-17.



’232 Patent, FIG. 2 (annotated)

39. The ’232 patent explains that “low-level information” includes, for instance, “sensed data” collected by sensor networks, and the higher-level information may include a notification that temperature readings exceed a threshold, and a report indicating a probable cause or situation associated with the notification (e.g., “high probability of problem in the heating machine”). ‘232 Patent, 6:1-33 (Table 1).

40. As shown below, all the above sensor data analysis concepts, including a sensor network middleware performing the analysis to generate conditional events, context-aware information, and other knowledge contents were previously known in the art.

VII. CLAIM CONSTRUCTION

41. It is my understanding that in order to properly evaluate the '232 patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry. I have reviewed the entirety of the '232 patent, as well as its prosecution history. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability of the Challenged Claims, no claim terms require express construction.

42. I have also been informed that in the district court proceeding concerning the '232 patent, the Patent Owner's position in its Infringement Contentions is that no terms invoke §112, ¶6.

VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

43. I have been asked to provide my opinion as to whether the Challenged Claims of the '232 patent would have been obvious in view of the prior art. The discussion in this Declaration provides a detailed analysis of how the asserted prior art references teach each limitation of the Challenged Claims.

44. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

45. As described in detail below, it is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA before the time of the alleged invention of the '232 patent. I apply the prior art as follows:

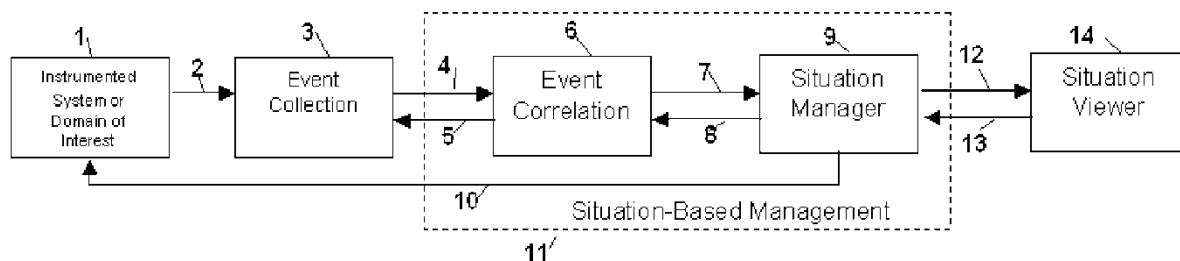
Grounds	Claim(s)	Basis
#1	1-12	§103 Jakobson and Tsetsos

A. Ground 1: Claims 1-12 are obvious under §103 over Jakobson in view of Tsetsos.

1. Summary of Jakobson

46. Jakobson describes methods and systems for “context-sensitive event correlation” and “situation-based management” using data collected from “a wide variety of types of sensors.” Jakobson, Abstract, [0015]. Jakobson’s situation-

based management (SBM) techniques include collecting and processing sensor data from a domain to identify higher-level “events” and “situations.” FIG. 1 of Jakobson shows an example situation management process flow including functional modules for collecting sensor data and identifying situations.



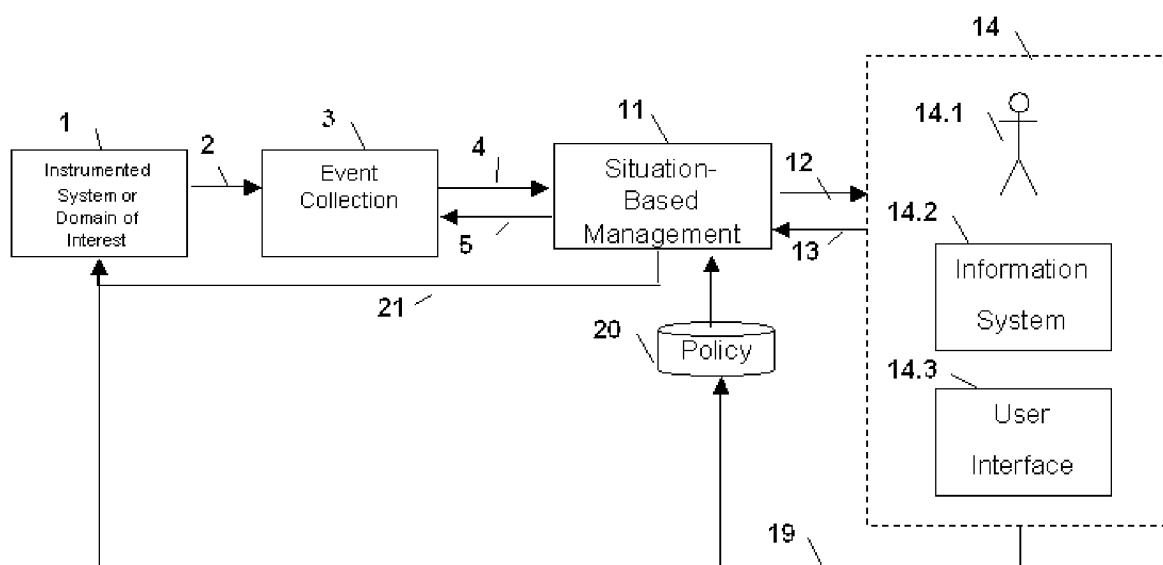
Jakobson, FIG. 1

47. These “correlated” events and situations describe more general conditions of the domain compared to basic sensor readings. *See* Jakobson, [0019] (e.g., a “root-cause”), For instance, a collection of low-level sensor data representing sound, pressure, temperature can be combined, correlated, and processed to identify conditions like “power disruption,” or “chemical release.” Jakobson, [0010]-[0013], [0177]. The SBM situations are provided to users through a “situation viewer” program that may employ a user interface or information system to view and respond to situation reports:

FIG. 4. shows that the situation viewer 14 may be a human observer 14.1, an information system 14.2, or a user interface 14.3. A human observer 14.1 may be provided situations, situation reports, extracts of situations, abstracts or summaries of situations, or descriptions of

situations through communications channel 12 such as a user interface, an instant message, an email, and a voice, image, or video input system. An information system 14.2 may interface to situation-based management 11 through various means for distributed information systems to exchange information such as messages, remote procedure calls, and shared databases.

Jakobson, [0178].



Jakobson, FIG. 4

48. Jakobson refers to a broad range of sensor-based information as “events.” Jakobson, [0012]. Events can include raw sensor data, filtered and consolidated sensor readings, and higher-level descriptions of the state of the domain. Jakobson, [0012], [0013], [0019]. Jakobson explains that the low-level “events” provided by the sensor networks (e.g., raw or filtered sensor data) are collected and forwarded to a “Situation-Based Management” (SBM) module. The

SBM module uses an iterative series of signal fusion, data fusion, and event correlation processes to infer higher-level “correlated” events. Jakobson describes implementing SBM systems using “middleware,” which facilitates “the building of open, scalable, and customizable systems.” Jakobson, [0215], [0216].

49. Signal fusion operations include:

...extracting information from the signals, filtering the signals, converting the signals to digital format, eliminating redundant information, and combining multiple signals including historical signals into visual or audio patterns.

Jakobson, [0010].

50. Data fusion operations include:

filtering, translating, cleansing, aliasing, aggregating, and converting to a uniform representation.

Jakobson, [0013].

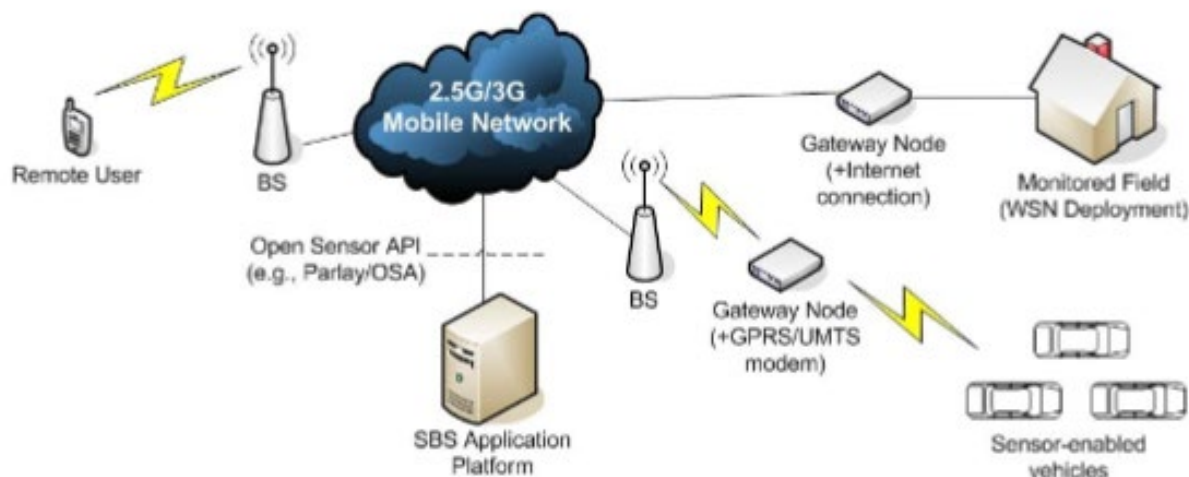
51. The event correlation services perform event correlation operations on the fused data from the signal and data fusion operations. Jakobson, [0013]-[0014]. These event correlation operations include filtering, compressing, counting, and other conditional operations to generate the higher-level correlated events; represented as consolidated, filtered, and/or thresholded compilations of sensor information. Jakobson, [0187]-[0188].

52. The correlated events are provided to a “situation manager” submodule within the SBM module. Jakobson, [0022]-[0026]. The situation

manager compares the incoming correlated events to a collection of “situation templates” to determine whether the correlated events match a given situation with a certain degree of confidence. Jakobson, [0026], Abstract. The situation manager generates situation reports and summaries, including their associated context. Jakobson, [0178]. A “situation viewer” including an information system and a user interface receives situation reports and presents them to the user, who can then make decisions on how to address any detected situations. Jakobson, [0178]-[0180].

2. Summary of Tsetsos

53. Tsetsos relates to “sensor network” applications that facilitate smart, “context-aware” services. Tsetsos, Abstract, 1. Tsetsos describes using gateway devices, middleware, and application program interfaces (APIs) to manage heterogeneous sensor networks in a unified form. Tsetsos, 3 (gateway nodes are “the connecting link between the sensor network and the WAN”), 5 (discussing challenges resulting from “heterogeneity” of wireless sensor networks). FIG. 1 illustrates gateway nodes collecting and transmitting data from wireless sensor networks (WSNs) to the Internet.



Tsetsos, FIG. 1.

54. Tsetsos describes using middleware to provide “unified handling” of heterogeneous WSNs. Tsetsos, 5. Tsetsos proposes using known “industry standard[s]” like Parlay/OSA (Open Services and OSGI (Open Services Gateway Initiative) as the framework for the middleware. Tsetsos, 5. These protocols specify APIs that provide developers with a broad set of “network functionality (user interaction, mobility management, user terminal capabilities, location, etc.).” Tsetsos, 5.

3. Reasons to Combine Jakobson and Tsetsos

55. As an initial matter, a POSITA, when considering the teachings of Jakobson, would have also considered the teachings of Tsetsos. Both Jakobson and Tsetsos relate to middleware for collecting and analyzing sensor data from heterogeneous sensor networks and are therefore analogous art to the ’232 patent.

Jakobson, [0014]-[0015], [0215]-[0216]; Tsetsos, 5, 6, 8; ‘232 Patent, Abstract, 1:22-32.

56. Jakobson describes various service modules and mechanisms for the SBM middleware to interface and communicate with the sensor networks and the situation viewer (or situation monitoring clients in FIG. 7). For instance, Jakobson describes obtaining sensor data from a variety of sensors and using “protocol conversion functions” to provide the sensor data to the middleware in a suitable form. Jakobson, [0014]-[0015], [0216]. Further, the middleware includes services that control the presentation, form, and provisioning of situation management information to a situation viewer application (or situation monitoring client). Jakobson, [0177] (“situation may be represented in several formats”), [0178] (situation information “may be converted to a different format prior to or during the step of providing it to the information system”), [0218] (“Presentation 35 to support the user interface”), [0216] (“Event Notification provides facilities for subscribing to and filtering events”).

57. Jakobson’s situation manager and situation viewer communicate with and control the sensor networks through communication channels. Jakobson, [0175] (situation manager “may take action” on the domain of sensors through communications channel 10), [0180] (“situation viewer 14 may also operate on situation-based management 11 via communications channel 13”).

58. Jakobson, however, does not discuss at length the interfaces between the sensor networks, the middleware, and the situation viewer application, because such interfaces were well-known and include implementation details that Jakobson expects.

59. For example, Tsetsos also describes using middleware to communicate with heterogenous sensor networks, with additional implementation details for the interface and communications between sensor networks, middleware, and third-party applications. Tsetsos, 5, 6 (“a Sensor SCS/API, can provide **developers** with a common way for the interaction between the service logic and the actual sensor network”), FIGS. 3 and 4. These additional implementation details include a sensor network gateway node using well-known gateway protocols, and application program interfaces that facilitate unified, and flexible communications (and controls) between the sensor network, the middleware, and third party applications. Tsetsos, 3, 5-6.

60. As explained in the mapping below, a POSITA would have been motivated to use Tsetsos’s gateway and associated APIs in Jakobson’s SBM system to simplify and standardize interactions with heterogenous networks of sensors and with situation viewing and monitoring applications (e.g., user interface software, instant messaging application, email). Using these well-defined interfacing technologies would have made sensor data collection and situation

reporting more robust, efficient, and easily scalable. In that regard, Tsetsos simply provides middleware functionalities that Jakobson expects, and that would advance Jakobson's objectives of unifying diverse sensor data and formats, and providing scalable, fault-tolerant systems for situation management. Jakobson, [0014]-[0015], [0215]-[0216].

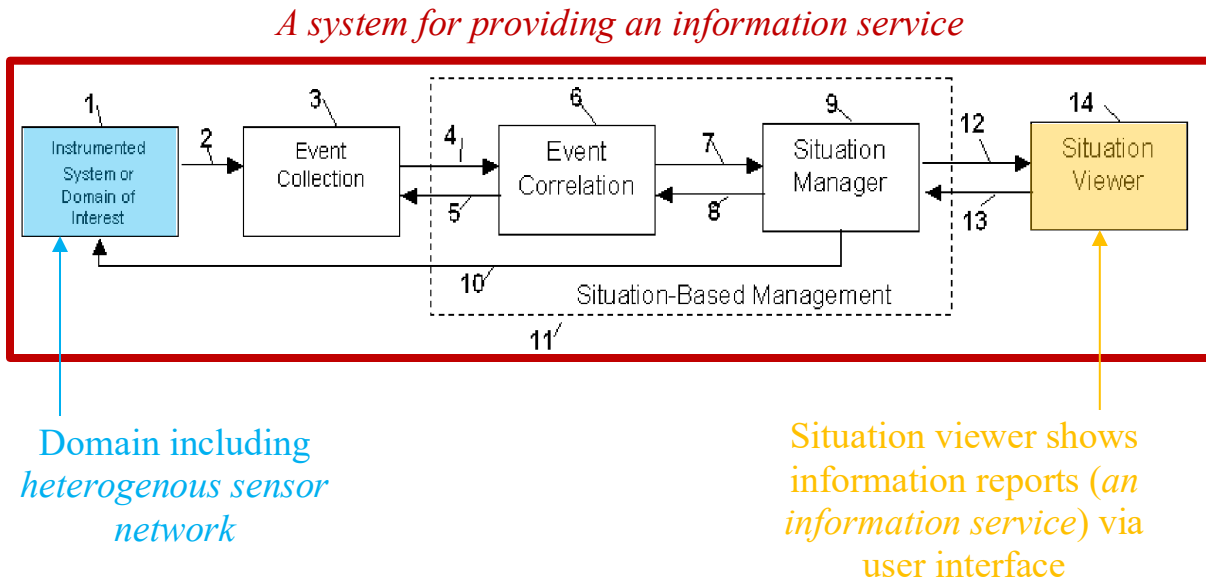
4. Detailed Analysis of the Claims

a) Claim 1

[1.0] *A system for providing an information service based on heterogeneous sensor networks, comprising:*

61. First, Jakobson discloses a *system* for situation-based management (SBM). Jakobson, Abstract, [0009]. Jakobson explains that its SBM receives data from “sensor networks” and outputs information reports to users to a situation viewer including a user interface. Jakobson, [0178].

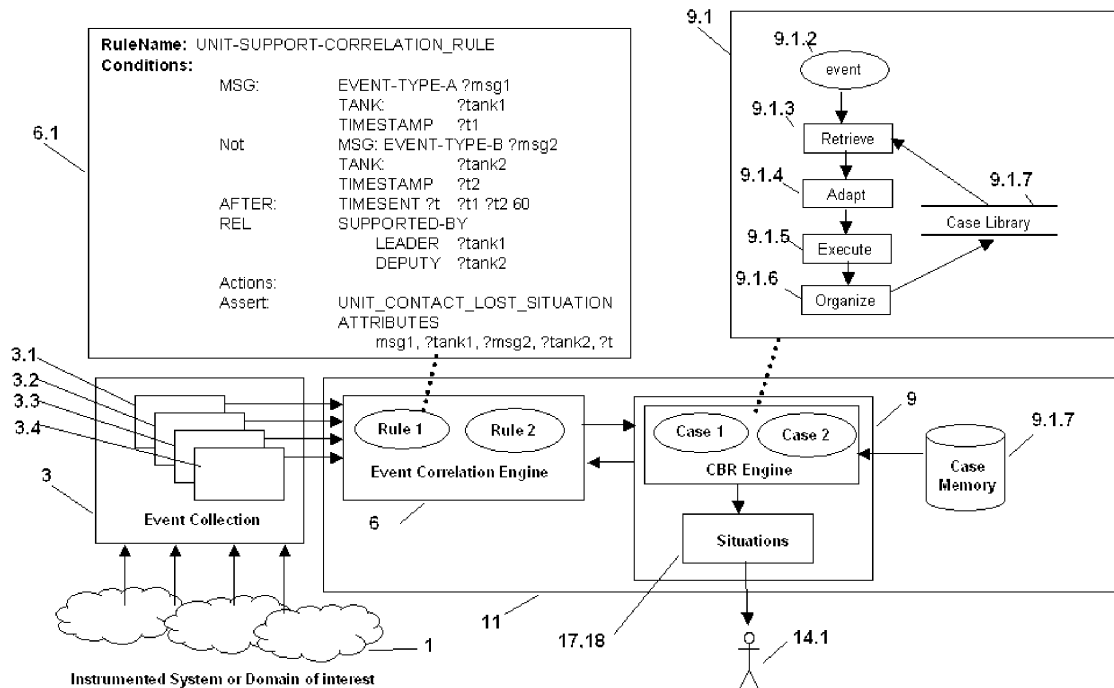
62. FIG. 1 shows a diagram of a method accomplished using a set of functional modules or blocks, including a “domain of interest” that is instrumented with sensors, an SBM component including an event correlation module 6 and a situation manager 9, and a situation viewer 14. Jakobson, [0174].



63. FIGS. 2-5 are similar diagrams illustrating SBM methods, each figure highlighting certain aspects of SBM with varying levels of detail. *See* Jakobson, [0138]-[0142] (Brief Descriptions of Figures for “a method of managing a domain” for “situation-based management”).

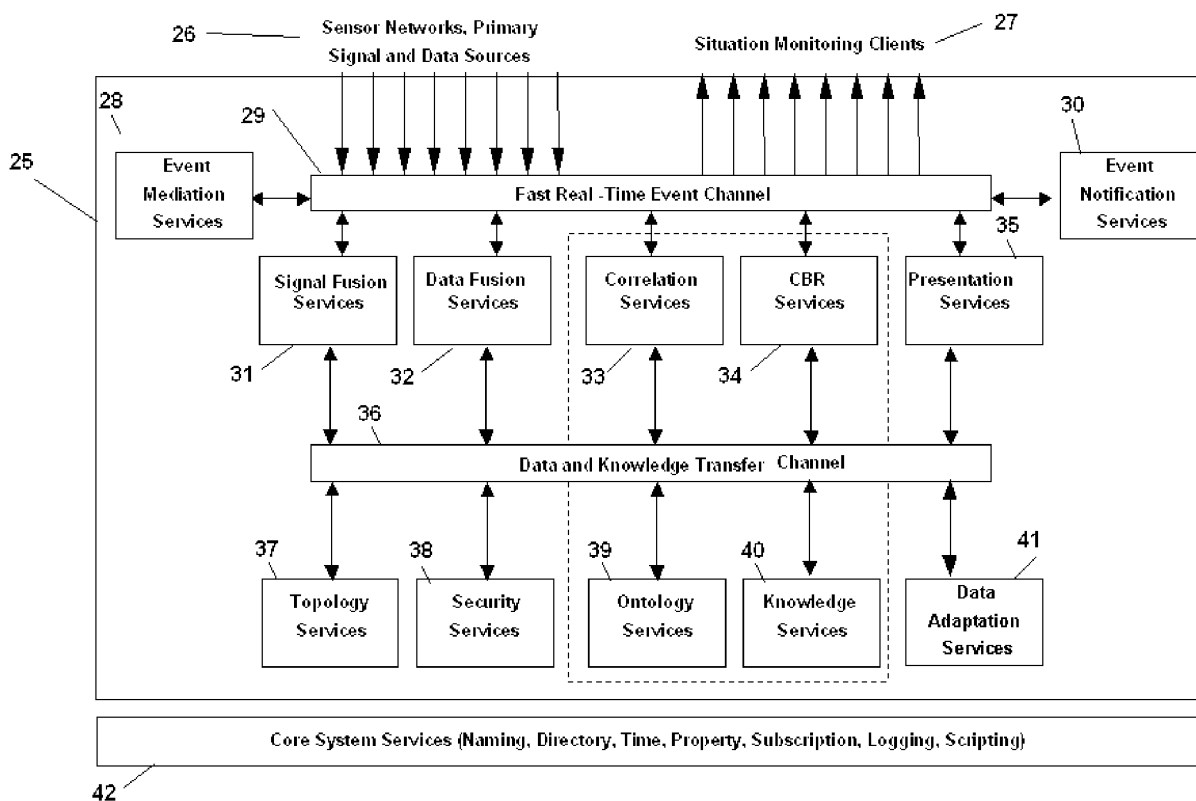
64. While FIGS. 1-5 are described as showing “methods,” FIGS. 6-8 of Jakobson illustrate “system[s]” for SBM according to a variety of software architectures. For instance, Jakobson’s FIG. 6 illustrates a system “for realizing SBM.” Jakobson, [0185]. Although Jakobson does not specify a software architecture for the system in FIG. 6, Jakobson explains that the system includes an event correlator that uses a “rule-based engine,” and a situation manager that uses “case-based reasoning.” Jakobson, [0209]. Jakobson also explains that “distributed

services architecture” including “middleware” can be used to implement an SBM system. Jakobson, [0215].



Jakobson, FIG. 6

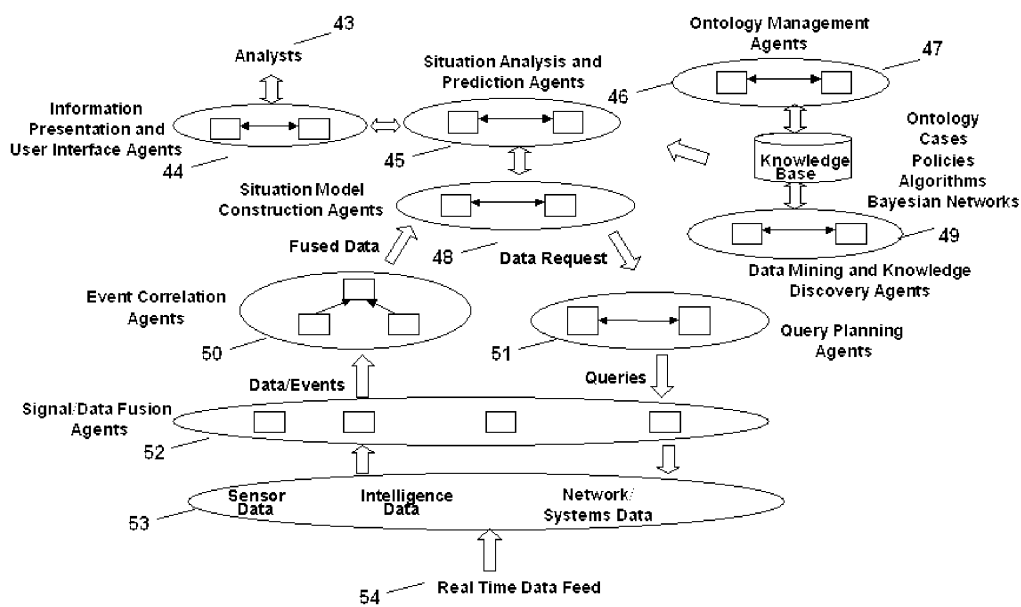
65. In that regard, Jakobson’s FIG. 7 illustrates a system for SBM using “a distributed services architecture.” Jakobson, [0144]. The distributed services architecture of FIG. 7 uses “middleware services 25” to realize the SMB. Jakobson, [0216].



Jakobson, FIG. 7

66. FIG. 8 shows a system for SBM “using an agent-based architecture.”

Jakobson, [0145], [0221].



Jakobson, FIG. 8

67. Software architectures like the distributed services (including middleware) and agent-based architecture would have been background knowledge to a POSITA for developing a software providing an information service. *See* Newton, 526; Tanenbaum, 2 (describing middleware). Thus, Jakobson's middleware and distributed services frameworks are simply implementation details that would have been selected based on the developer's preferences and resources. A POSITA would have understood that the functionalities provided by Jakobson's SBM system could be achieved through several different software architectures or frameworks, including the architectures Jakobson describes. For instance, the embodiments illustrated in each of FIGS. 6-8 of Jakobson include the same functional modules (e.g., data fusion 3.3 (FIG. 6, [0181]), data fusion services 32

(FIG. 7, [0216]), data fusion agent 52 (FIG. 8, [0221])). Indeed, the same functional modules are also represented and discussed in FIGS. 1-5. Jakobson acknowledges that such software architectures were well-known and “standard.” *See* Jakobson, [0215] (distributed software architectures use “**standard** middleware services...and **standard** inter-component communication protocols.”), [0220] (“in agent-based software architecture, system function is distributed into modules of...software objects called agents,” using known agent-based software protocols such as ADK and LEAP.).

68. Thus, the embodiments of FIGS. 6-8 are discussed in the context of different software architectures, but the functionalities in the SBM modules they describe are the same. The disclosure for each figure may highlight different aspects of the same SBM modules. To the extent this Petition refers to the teachings of FIGS. 6-8, this Petition does not necessarily require any of the specific software architectures, nor does this Petition suggest a combination of the separate embodiments or their respective software frameworks. For instance, Jakobson’s teachings regarding the functionalities of “data fusion” module in FIG. 6 and the data fusion agent 52 in FIG. 8 also apply to the “data fusion services 32” illustrated in FIG. 7. The implementation of the data fusion module as a distributed services object, a software agent, or any other software-based functional object does not affect the analysis below.

69. Second, Jakobson explains that situation-based management (SBM) includes outputting reports, notifications, and other information (*information service*) to a user via a “situation viewer 14” that includes “an information system 14.2 or a user interface 14.3.” Jakobson, [0178]. For instance, Jakobson explains, referring to FIG. 4:

A human observer 14.1 may be provided situations, **situation reports, extracts of situations**, abstracts or summaries of situations, or descriptions of situations through communications channel 12 such as **a user interface**, an instant message, an email, and a voice, image, or video input system.

Jakobson, [0178].

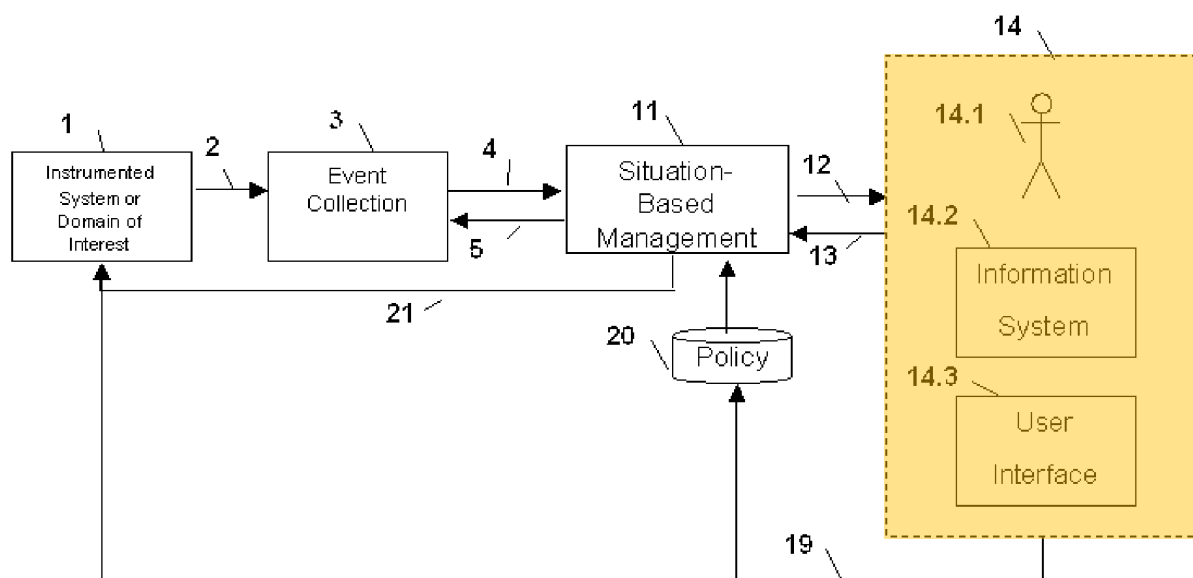


Fig. 4

Jakobson, FIG. 4 (annotated)

70. Jakobson explains that the situation viewer 14 may also execute operations on the situation management module 11, including “changing properties of situations, creating new situations, modifying the knowledge base of situation-based management, and modifying the ontology of the situation manager.” Jakobson, [0180].

71. Thus, Jakobson’s SBM system, including the situation viewer 14 which provides situation reporting and situation management operations, discloses *a system for providing an information service*.

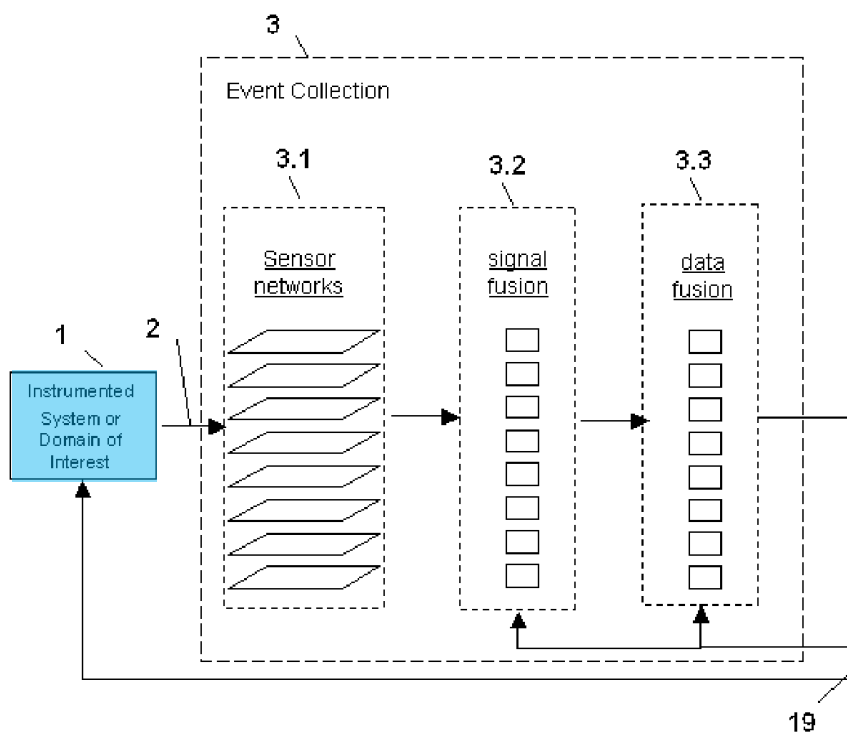
72. **Third**, referring to FIG. 6, Jakobson explains that SBM system receives sensor information from a “domain” having “**sensors and agents** which communication through **sensor networks**.” Jakobson, [0181]. The sensors may provide data “in digital or analog format.” Jakobson, [0181]. Jakobson explains that the sensor data are provided by “**a wide variety of types of sensors** to be incorporated into a situation-based management system.” Jakobson, [0014]. These various types of sensors represent a variety of environmental conditions and characteristics, such as “audio, pressure, or temperature.” Jakobson, [0010]. Thus, Jakobson’s system for SBM (*system for providing an information service*) is based on *heterogeneous sensor networks*.

73. Accordingly, to the extent the preamble is limiting, Jakobson discloses it.

[1.1] a sensor node constructing a sensor network to sense environment information

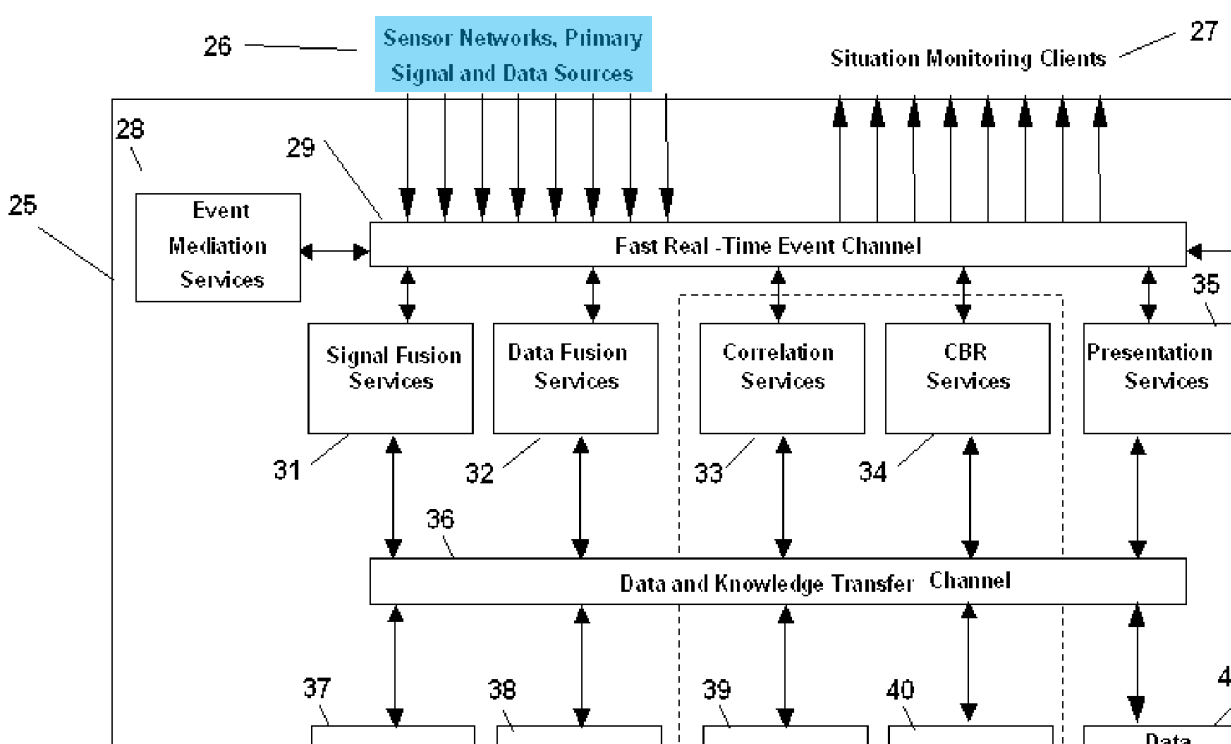
74. First, Jakobson's "domain 1 is instrumented with sensors and agents which communicate through sensor networks to provide sensor information."

Jakobson, [0181]. Each sensor instrumented at the domain is *a sensor node*. FIG. 5 of Jakobson shows the domain instrumented with sensors (blue).



Jakobson, FIG. 5 (partial)

75. Consistent with the description of FIG. 5, Jakobson's FIG. 7 illustrates a system "[f]or realizing SBM using middleware services." Jakobson, [0216]. The system includes "sensor networks" (blue) which provide sensor data.



Jakobson, FIG. 7 (partial annotated)

76. The *sensor nodes* instrumented at the domain form or provide a sensor network, which discloses or renders obvious “a *sensor node constructing a sensor network*.” For instance, it would have been obvious for one or more of the sensors of the domain to communicate with one or more other sensors, a data processing system (e.g., Middleware services 25 in FIG. 7), or a sensor gateway device (e.g., gateway node of Tsetsos, discussed below for limitation [1.2]), thereby forming a *network*.

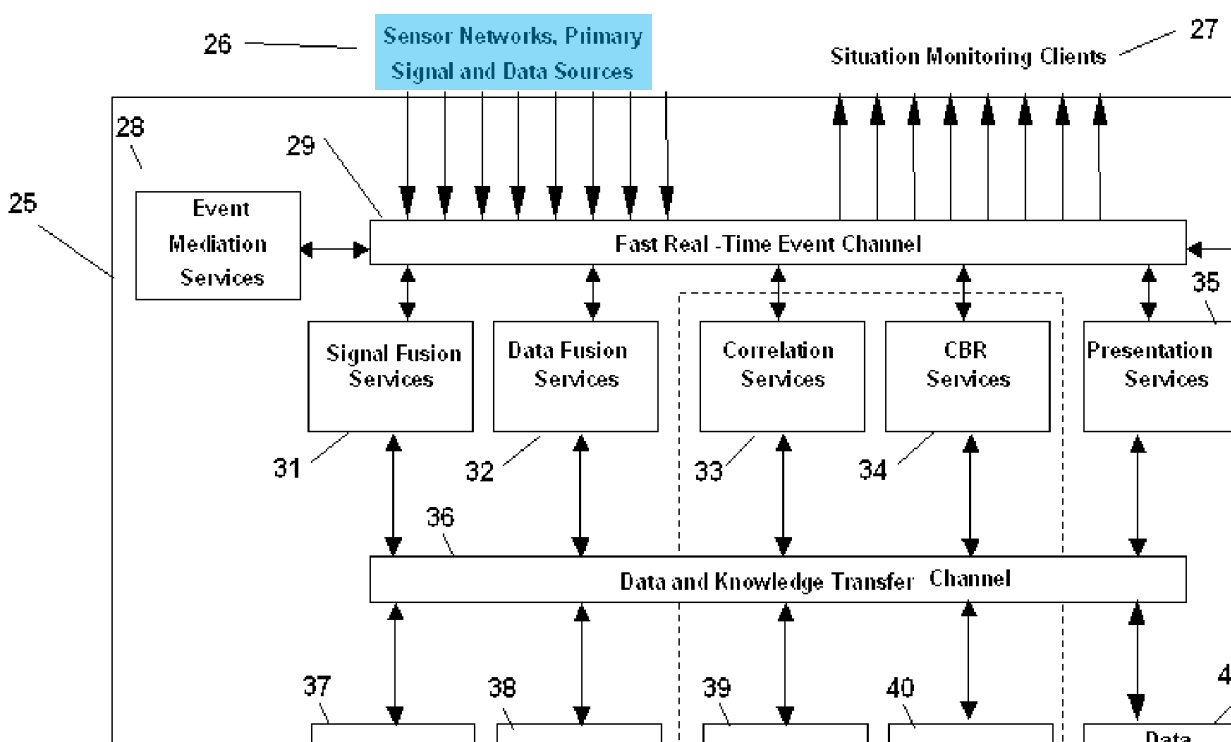
77. **Second**, Jakobson explains that sensors collect *environment information* from “real-time sources including sensors on field equipment, air

surveillance, field personnel, satellite, and ground radar.” Jakobson, [0007]. The information collected from the sensors represents, for instance, “audio, pressure, or temperature” (*environment information*). Jakobson, [0010].

78. Thus, Jakobson discloses or at least renders obvious the *sensor network* sensing *environment information*, alone or in view of Tsetsos.

[1.2] *a sensor network data transmitter for collecting sensed data sensed by the sensor in real-time and transmitting the collected sensed data in a message form; and*

79. As discussed above in [1.1], Jakobson’s FIG. 7 illustrates that its middleware 25, receives sensor data from “sensor networks” (in blue) and transmits the sensor data to the signal and data fusion services via the “Fast Real-Time Event Channel 29.” Jakobson, [0216], [0219].



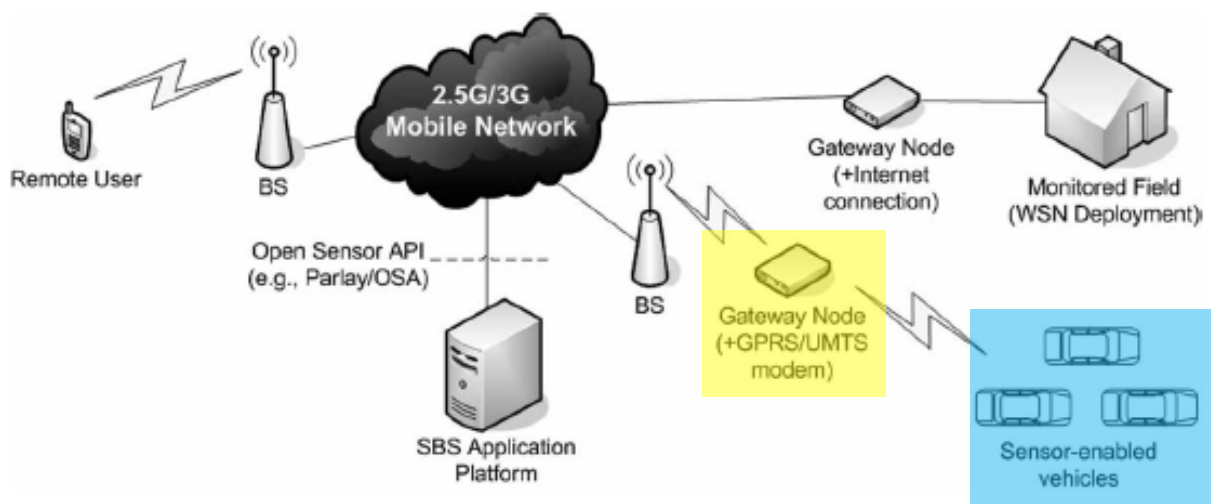
Jakobson, FIG. 7 (partial annotated)

80. To the extent Jakobson doesn't explicitly disclose that the sensor networks transmit data to the middleware via a "*sensor network data transmitter*," using an intermediate module or node to collect, aggregate, and transmit data was well known and commonly-used. This would have been a straightforward and predictable inclusion of well-known techniques and devices in Jakobson's system to achieve the predictable results of routing sensor data through a gateway node. *See supra*, § V.B.

81. For instance, Tsetsos describes that typical modern sensor networks utilize a "gateway node" which operates as the "connecting link between the

sensor network and the WAN (i.e., mobile network or internet).” Tsetsos, 3. In

FIG. 1, Tsetsos illustrates how the Gateway Node (yellow) links a domain of sensor-enabled vehicles (blue) to a mobile network.



Tsetsos, Fig. 1.

82. In one implementation, Tsetsos explains that a Gateway node uses an Open Services Gateway Initiative (OSGi) protocol to aggregate sensor data and provide the aggregated sensor data in a standardized format:

The deployment of the WSN driver in **an OSGi Gateway** located in the monitored field enables **the aggregation of the sensor data** and thus is more cost- and bandwidth-efficient. Additionally, OSGi fully supports the dynamic management of the WSN driver (e.g., remote installation of updates) and has been **adopted by major industries (e.g., for vehicular telematics and home automation solutions).**

Tsetsos, 7.

83. Tsetsos explains that the gateway “has more processing, energy and communication capabilities than the other nodes and is the connecting link between the sensor network and a WAN (i.e., mobile network or Internet).”

Tsetsos, 3.

84. It would have been thus obvious for Jakobson’s instrumented domain of sensors to include, or communicate through, a gateway like Tsetsos’s gateway to leverage the increased computational power and communication capabilities. Gateway devices like the one described in Tsetsos were well-known and relied on standardized communication protocols to be easily and scalably integrated into a sensor network. Using a sensor gateway would have reduced the hardware and software requirements on the individual sensors, and allowed for more efficient communication of aggregated sensor data from the various sensor nodes to the middleware services. *See* Tsetsos, 3. Tsetsos also teaches that “[t]he gateway node has more processing, energy and communication capabilities than the other nodes and is the connecting link between the sensor network and a WAN (i.e., mobile network or Internet).” *See* Tsetsos, 6. A POSITA would have recognized that a gateway node would have longer range communication capability than the sensor nodes, and thus allow for wider area deployment of sensor nodes without requiring the sensor nodes to have similar increased capability.

85. A POSITA would have recognized the advantages of funneling heterogenous sensor data through a gateway device designed for that purpose, instead of managing several different sensor communication pathways, with perhaps several different proprietary communications protocols and/or data formats. For instance, a gateway like the one in Tsetsos could simplify the process of adding or removing sensors from the sensor domain. *See* Tsetsos, 2 (wireless sensor networks frequently change and adapt). This provides benefits that are analogous to those provided by Jakobson’s protocol conversion functions. *See* Jakobson, [0014]-[0015] (object of event collection is “to permit: a wide variety of types of sensors to be incorporated into a situation-based management system”), [0215] (distributed software architecture facilitates “open, scalable, and customizable systems”). Indeed, a POSITA would have understood that the object of middleware and distributed software architectures is to facilitate communication and translation between heterogeneous system components that use different protocols and data formats. *See* Newton, 526 (“middleware serves as a translation mechanism, gluing together application software across a network”); Tsetsos, 5 (middleware provides “unified handling” of heterogeneous networks), 6 (middleware supports “ubiquitous context-aware services”).

86. A POSITA would have had a reasonable expectation of success in combining Tsetsos’s gateway device and its associated protocols, since gateways

were well-known and used industry standard protocols (e.g., OSGi), as explained above. Incorporating Tsetsos's gateway would have involved straightforward networking and computer programming techniques to allow Jakobson's SBM system to receive messages from the gateway. For the same reasons, the results of the combination would have been predictable.

87. In the combination, the gateway would *collect[]* the sensor data and *transmit[] the collected sensed data*.

88. **Second**, Jakobson explains that the data received from the sensors is "real-time data." See Jakobson, [0221], FIG. 8 (describing "**Real Time** Data Feed" 54). Thus, the data collected from the sensors is collected and provided *in real-time*. See also, Jakobson, [0216], [0219]. As explained above, it would have been obvious for the sensor data to be collected from various sensors and forwarded to the sensor networks 3.1 using the gateway described in Tsetsos. Because Jakobson describes that the sensor data is collected in real time, it would have been obvious for the gateway (*sensor network data transmitter*) to similarly collect and forward the sensed data in *real time*, to ensure that the SBM services can analyze and respond to events and situations identified from the sensed data.

89. **Third**, Jakobson refers to an event, including an event received from the sensor networks, as "*a message*," and notes that events can be represented in forms like "**human-readable text**, digital images or audio or video, digital

measurements of signals, and positions of objects in a global positioning system.”

Jakobson, [0011]-[0012]. In that regard, Jakobson teaches that its middleware receives the sensor data as “events”:

The sensors and signal fusion equipment directly **provide events about the domain over one or more data networks to one or more data processing systems** or the events may be stored for collection at sensors, signal fusion equipment or other data processing equipment. Jakobson, [0011]; *see also*, [0175] (events are “provided from the domain’s instrumentation....”).

90. Thus, Jakobson describes communicating sensed data as a *message*, and also explains that the inter-component SBM modules communicate information as a *message*. In that regard, Jakobson explains that its event collection and event distribution mechanisms rely on communicating data via “messages:”

The object of the event collection and event distribution mechanism in situation-based management is to...distribute event processing so that **large volumes of messages can be efficiently processed**. Jakobson, [0014]-[0016].

91. Communicating data in message form was widely known and used, as “messages” simply refers to “a sequence of characters used to convey information or data.” Newton, 517. Thus, Jakobson discloses or renders obvious that the sensor networks provide an event *in a message form*.

92. Additionally, it would have been obvious for the gateway (*sensor network data transmitter*) to *collect* the events from the sensors and to *transmit* the collected events *in a message form*, consistent with Jakobson’s approach for handling events provided by the instrumented domain. As explained above, Tsetsos explains that the gateway aggregates sensor data and transmits the data via the Internet. Tsetsos, 3, 7. Data transmitted via the internet is sent in IP packet form (“*message form*”). For instance, Newton’s Telecom Dictionary explains that, in the TCP/IP protocol, which is the standard communication protocol for Internet communications, transmits “messages” in the form of “packets.” Newton, 812-813. In that regard, the ’232 patent refers to the well-known TCP/IP communication protocols for communicating sensor data, acknowledging that this protocol was well-known and understood. Newton also explains that “Internet Protocol (IP)” is used for “rout[ing] outgoing **messages**, and recogniz[ing] incoming **messages**.” *See also*, Newton, 611-612 (when packets arrive at destination, their headers are detached and discarded to obtain the payload).

93. A POSITA would have known that when packets from an external network come in, the packet’s headers are stripped off as the packet’s payload (data) is extracted by the upper layers. Tanenbaum, 41-44; Newton, 517-518, 611-612.

94. Additionally, Tsetsos also describes the use of standard protocols and application programming interfaces (APIs) like Parlay/OSA (Open Services Access) to facilitate sensor data retrieval:

The accumulated experience from the wireless networking applications dictates **the standardization and adherence to open public APIs for the interaction between the applications and the network elements**. Since the WSN can be regarded as a network element, the support of de facto standards for the WSN handling (i.e., **sensor data retrieval**) by the platform seems both crucial and feasible. Such **an API could be the Parlay/OSA (Open Services Access)**.

Tsetsos, 3.

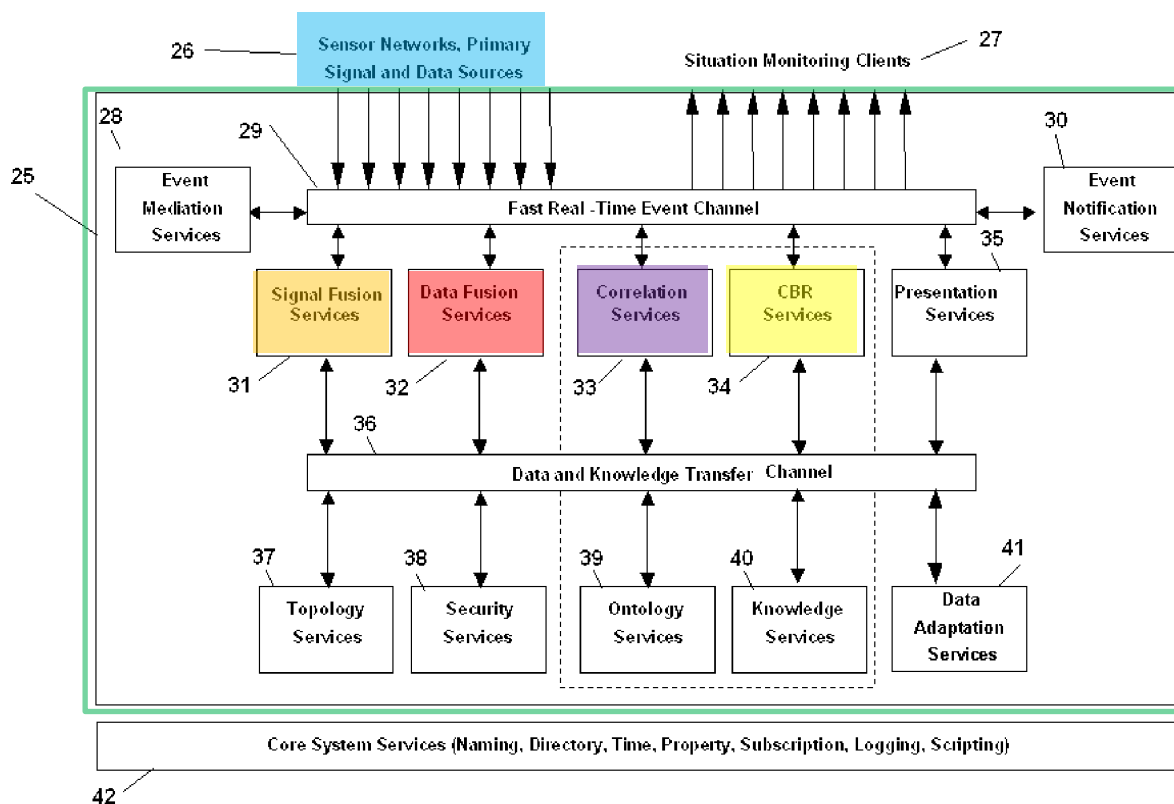
95. It would have been obvious to use a standard message format for transmitting the aggregated data (e.g., as IP packets, API requests and responses) to improve the reliability and robustness of the data communication. For instance, a POSITA would have relied on well-established and developed digital communication protocols (e.g., the Internet Protocol (IP), 3GPP protocols, IEEE 802.11 (Wi-Fi)), which communicate data *in a message form*, and which include refined error prevention and error management features to maintain reliable connections. Additionally, it would have been obvious to use APIs to harmonize interactions with heterogeneous sensors to allow developers to provide a scalable way to control or retrieve sensor data without having to accommodate every sensor's specific protocol and data format.

96. In the event this limitation (or any other limitation) is construed as a means-plus-function limitation, Jakobson's general purpose "computer" executing a "software architecture" that implements "middleware services or components with well-defined functionality" carries out the functions recited in association with this limitation (and all other limitations in the Challenged Claims). *See* Jakobson, [0280] (operations "realized on a multiprocessor computer"); [0215] (SBM middleware realized by software architecture); *see also* Tsetsos, 3 (gateway nodes with processing and communication capabilities), 5 (mediation software based on Open Services Gateway Initiative).

97. Thus, Jakobson discloses this limitation, or renders it obvious alone or in view of Tsetsos.

[1.3] *a Ubiquitous Sensor Network (USN) middleware for extracting object and environment information by cleaning, classifying and integrating the message received from the sensor network data transmitter,*

98. **First**, Jakobson discloses "middleware services" 25 (green box) for realizing situation based management (SBM). Jakobson, [0216].



Jakobson, FIG. 7 (annotated)

99. The middleware includes signal fusion services (orange), data fusion services (red), correlation services (purple), and case-based reasoning situation manager services (yellow) that use the data provided from the sensor networks (blue).

100. As explained above at § V.A, “Ubiquitous” sensor networks are those that collect real-time data from a variety of sensors, which Jakobson discloses. *See* Jakobson, [0009], [0010], [0014].

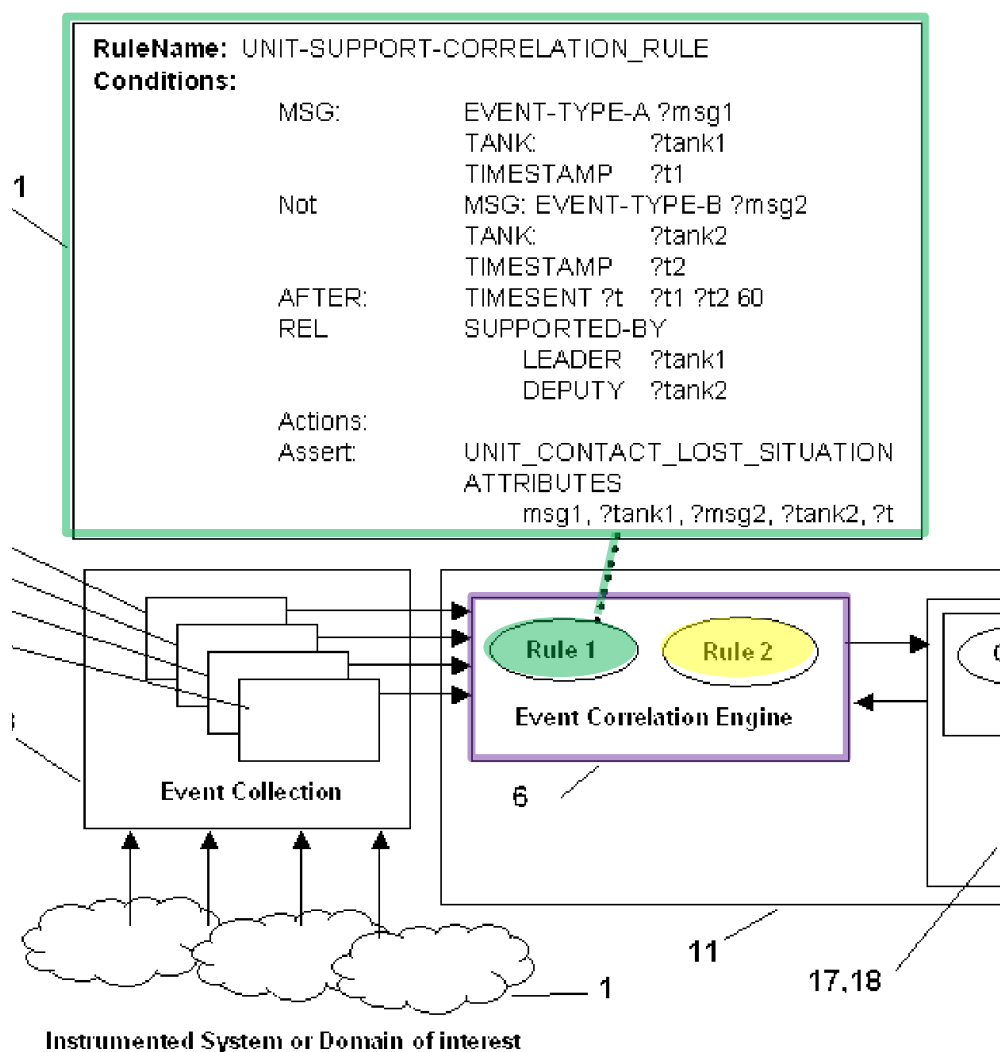
101. Jakobson's middleware services 25 discloses the "*Ubiquitous Sensor Network (USN) middleware*."

102. Jakobson explains that data fusion (red) operations include "filtering, translating, cleansing, aliasing, aggregating, and converting to a uniform representation." Jakobson, [0013]. Jakobson's event correlation service "**extracts** **the essential properties of the events** collected by event collection 3." Jakobson, [0175]. Additionally, the information output to the situation viewer 14 for the user includes "**extracts** of situations." Jakobson, [0178]. The extracted "properties" include *object and environment information* because the properties represent a condition of the domain and are based on the sensed data from the domain. Jakobson, [0170]-[0173], FIG. 3.

103. Thus, Jakobson discloses that the middleware 25 (*USN middleware*) *extract[s] object and environment information*.

104. Second, Jakobson explains that the data fusion and event correlation operations include *cleaning, classifying, and integrating* operations. For instance, Jakobson explains that "[d]ata fusion operates on the sensor data to **clean** the data, **alias** the data, **convert** the data to a uniform format, to **aggregate** the data, to persist the data, and other data fusion operations." Jakobson, [0182], [0013]. The fused data from data fusion services are provided to the event correlation module. Jakobson, [0013]. Event Correlation module 6 uses a rules engine to execute

various correlation rules or processes, such as event compression, filtering, and counting. These rules (green and yellow below) are algorithms instantiated using a known software framework called CLIPS:



Jakobson, FIG. 6 (partial annotated)

The event correlation engine 6 may use a general-purpose rules engine such as CLIPS (**C Language Integrated Production System**) to execute the correlation rules.

The types of rules used by the event correlation engine 6 include:

- (1) $[a, a, \dots, a] \rightarrow [a]$ Compression
- (2) $[a, p(a) \leq H] \rightarrow [\text{nil}]$ Filtering
- (3) $[a, C] \rightarrow [\text{nil}]$ Suppression
- (4) $[n \times a] \rightarrow [b]$ Counting
- (5) $[n \times a, p(a)] \rightarrow [a, p'(a), p' \geq p]$ Escalation
- (6) $[a, \text{subset}(a, b)] \rightarrow [b]$ Generalization
- (7) $[a, \text{subset}(b, a)] \rightarrow [b]$ Specialization
- (8) $[a \text{ T } b] \rightarrow [c]$ Temporality
- (9) $[a, b, \dots, T, \text{and}, \text{or}, \text{not}] \rightarrow [c]$ Logic/Temporality

Jakobson, [0185]-[0187].

105. As explained above with respect to limitation [1.0], Jakobson explains that “distributed systems” including “standard middleware services or components” are a common approach to software architecture. Jakobson, [0215]. It would have been background knowledge to a POSITA, and a matter of implementation using well-understood software architecture, for the rules-based event correlation services in FIG. 6 to be implemented using *middleware* services.

106. As explained below, these steps performed by the data fusion and event correlation modules also disclose or render obvious the *cleaning*, *classifying*, and *integrating* steps performed by the *USN middleware* of limitation [1.3].

“cleaning”

107. In a **first example**, Jakobson’s data fusion operations include “filtering” and “cleansing” the data provided from the sensors. Jakobson, [0013], [0182]. Filtering or cleansing the data discloses “*cleaning...the sensed data*.”

108. In a **second example**, Jakobson explains that the event filtering rule (2) involves discarding events that do not fall within a set of “predefined legitimate values”:

Event filtering (2) is the most widely used operation to reduce the number of events. If some parameter $p(a)$ of event a , e.g., priority, type, location, time stamp, etc., does not fall into the set of predefined legitimate values H , **then event a is simply discarded** or provided to a log file. The decision to **filter event a out or not is based solely on the specific characteristics of event a** . In more sophisticated cases, set H could be dynamic and depend on user-specified criteria or criteria calculated by the system.

Jakobson, [0187].

109. Event filtering discloses or renders obvious “*cleaning*” the events (*sensed data in the message*).

“classifying”

110. In a **first example**, Jakobson’s data fusion operations include “**data pattern recognition**” and “tracking object movements.” Jakobson, [0013], [0182]. Each of these data fusion operations discloses or renders obvious “*classifying*” the

sensed data because recognizing patterns or tracking objects from sensor data requires identifying distinguishing features and grouping similar data points, which includes classifying the sensor data into categories.

111. In a **second example**, Jakobson explains that the event correlation operations include “Event escalation (5)” which assigns values to events based on severity:

Event escalation (5) assigns a higher value to some parameter $p(a)$ of event a , usually the severity, depending on the operational context, e.g., the number of occurrences of the event.

Jakobson, [0188].

112. The event escalation including priority assignment is an example of *classifying* the sensed data, because this operation involves sorting and grouping events based on a parameter (e.g., priority).

113. Jakobson’s event correlation operations also include event generalization and event specialization, which organize events into a general class, or a more specific subclass, respectively:

Event generalization (6) is a correlation in which **event a is replaced by its super class b** . Event generalization has high utility for situation management. It allows one to deviate from a low-level perspective of events and view situations from a higher level. Event specialization (7) is an opposite procedure to event generalization. It **substitutes an event with a more specific subclass of this event**.

Jakobson, [0188].

114. Event generalization and specialization are also examples of *classifying* the sensed data, because they assign events to a “super class” or a “subclass,” respectively.

115. Further, claim 19 of Jakobson recites an “apparatus for providing context-sensitive information to external resources in situation-based management,” which includes “means to **classify information** as at least one of: incomplete information; inconsistent information,....” Jakobson, claim 19.

116. Thus, Jakobson discloses that the data fusion module, the event correlation module, or both, perform *classifying* of the sensed data (*the message*).

“integrating the message”

117. In a **first example**, Jakobson explains that the data fusion operations include “**aggregat[ing] the data**” to get the data (events) ready for processing by the event correlation module. Jakobson, [0013], [0182]. Jakobson further explains that the data fusion module “convert[s] the data to **a uniform format.**” Jakobson, [0013]. Aggregating the data with a uniform format to prepare the data for event correlation discloses “*integrating*” the data received from the sensor network (*the message*).

118. In a **second example**, the event correlation combines the events (*integrating the message*) to infer new events:

In yet a third embodiment, Event Correlation 6 may perform a “pre-processor, decision maker, and event correlation” function in which the **total collection of events known by Event Correlation 6 are combined to infer a new event** or a new quality of an event.

Jakobson, [0307].

119. Alternatively, the event correlation module integrates the lower-level events (*sensed data* in the *message*) into higher-level events that are more suitable for detecting or inferring the presence of situations and their associated causes.

The object of the event collection and event distribution mechanism in situation-based management is...to insure that events contain **sufficiently high-level content** suitable for processing by the EC....

Jakobson, [0014].

[O]ne object of the event correlation engine in situation-based management is to reduce a collection of events to a root-cause or a **high-level inferred synthetic event**.

Jakobson, [0019].

120. Correlating lower-level events to infer higher-level events discloses *integrating the message*.

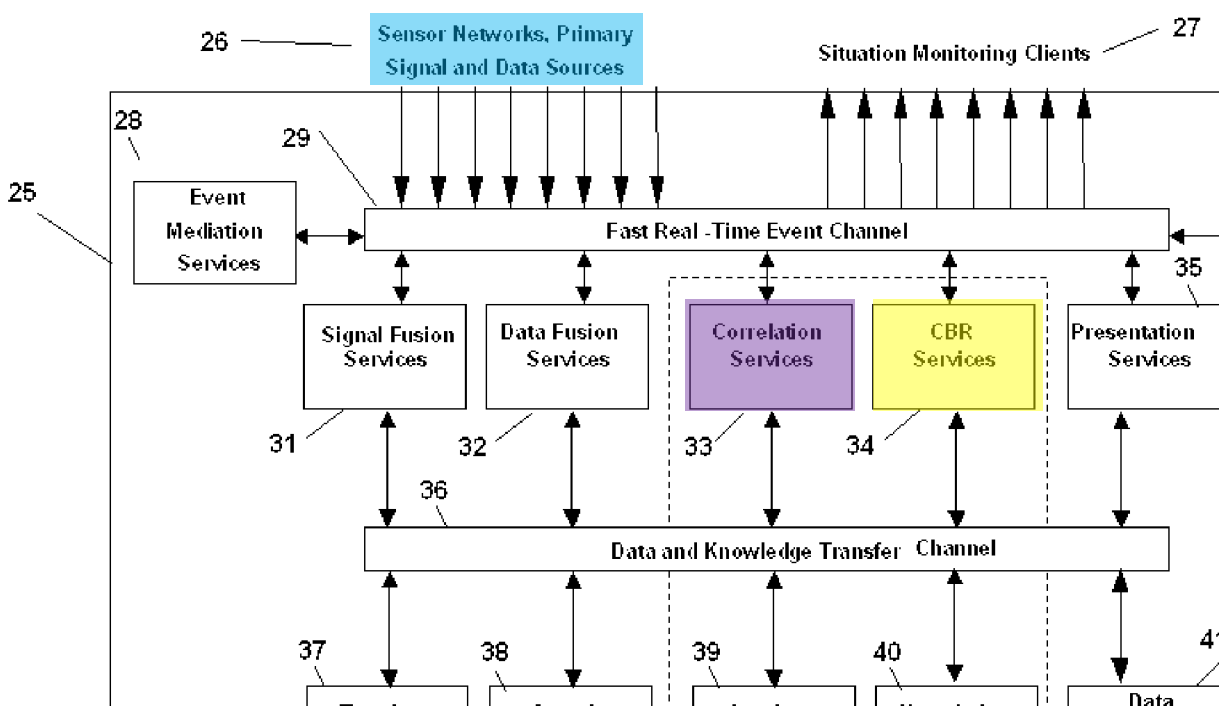
121. As explained above, Jakobson discloses that the result of the event correlation actions are to “**extract[] the essential properties of the events** collected by event collection 3.” Jakobson, [0175]. Thus, Jakobson discloses or at least renders obvious that the data fusion module, the event correlation module, or

both, *extract[] object and environment information by cleaning, classifying and integrating the events (the message[s])* provided from the sensor networks.

122. Thus, Jakobson discloses or renders obvious limitation [1.3].

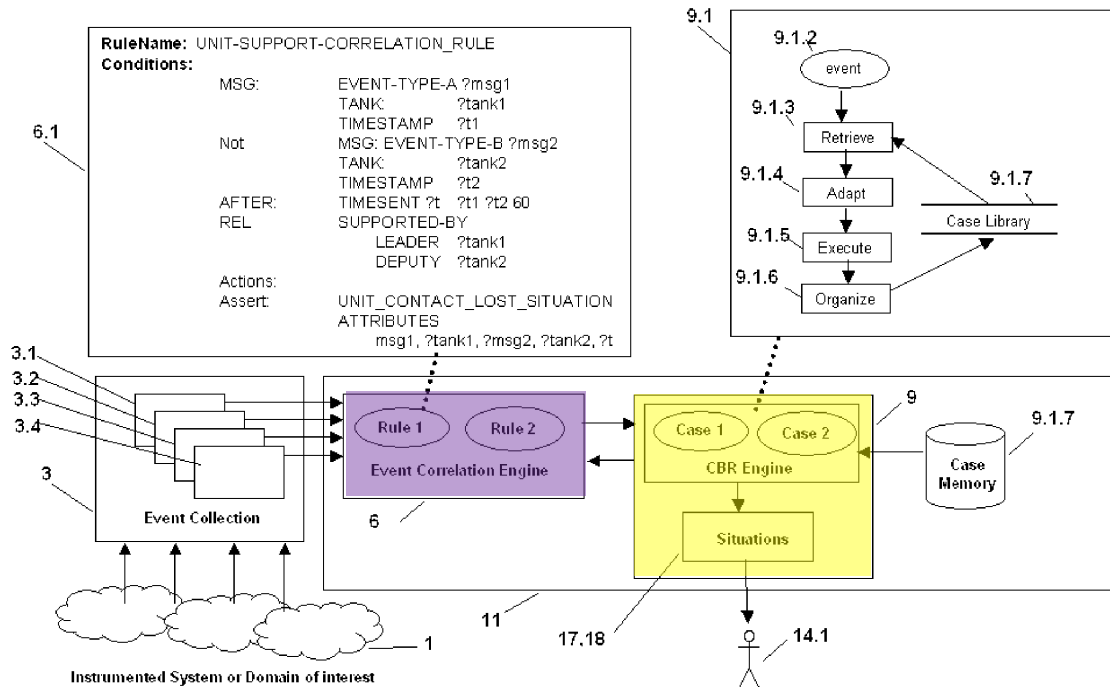
[1.4] *generating conditional events, context aware information, and knowledge contents from the extracted object and environment information, and*

123. Jakobson's SBM middleware (*USN middleware*) shown in FIG. 7 includes correlation services (purple) for performing event correlation tasks like filtering, compression, and escalation. Jakobson, [0217], [0219]. The SBM middleware also includes case-based reasoning (CBR) services (yellow) for situation management tasks, including creating and modifying situations, and providing contextual information associated with those situations. Jakobson, [0217], [0219].



Jakobson, FIG. 7 (partial annotated)

124. Similarly, Jakobson discloses that the SBM 11 includes event correlation modules 6 (purple) and a Situation manager 9 (yellow) that includes a CBR engine.



Jakobson, FIG. 6 (annotated)

“generating conditional events”

125. Jakobson’s event correlation module creates higher-level synthetic events based on lower-level events (e.g., sensor data) received from the sensor and data fusion modules:

For example, one object of the event correlation engine in situation-based management is to **reduce a collection of events to a root-cause or a high-level inferred synthetic event.**

Jakobson, [0019].

126. As explained above, Jakobson uses the term “event” to refer to a range of lower-level and higher-level information, from raw sensor data ([0181]) to detected complex patterns ([0188]), to “root-cause” synthetic events ([0019].

127. One type of such high-level inferred events involves “event clustering” that uses “Boolean operators and **conditional** (predicate) terms to create “complex correlation patterns.” Jakobson, [0188]. In a similar example, Jakobson explains that the event correlation 6 “tests the existence of the essential relations between” events, and the domain entities associated with the events, “evaluates the correlation predicates” based on those tests, and “generates of [sic] an event corresponding to [a] correlation predicate.” Jakobson, [0175]. A POSITA would have understood that the higher-level correlated events generated based on these tests and predicates are *conditional*, because they are conditioned or predicated upon application of tests or conditional terms.

128. In another example, Jakobson discloses the event correlation module creating high-level events based on an incoming event (*sensed data*) exceeding a threshold (*conditional*):

Events may contain modality such as “it is suspect that sensor 123 exceeded threshold.”

Jakobson, [0176].

129. Jakobson’s higher level inferred events are consistent with the ’232 patent’s description and examples of “conditional events,” which describe a detected event or condition, such as an average temperature falling below a threshold, as shown in Table 1 below (“low temperature: 13.5...for average temperature value”):

Classification	Analysis of sensed data and generation of information	Description
Conditional Events generated by USN middleware apparatus	Provide alarm ("low temperature: 13.5") for average temperature value	Application has already set a predetermined rule: alarm is generated to application program when average temperature for latest 1 hour is below 14° C.
Context aware information generated by USN middleware apparatus	Provide an imminent report about the given position ("There is high probability of problem in the heating machine; Check gas leakage or breakdown of the electricity")	USN middleware found the related rule and contents from its internal storage, as following. 1) There is high probability that problem will occur in heating facility when average temperature is below 14° C., current humidity is 80%, and it is night. 2) The latest problem occurring in the heating machine was related to the gas leakage and power supply failure.

'232 patent, Table 1 (partial annotated)

130. In another thresholding example, event correlation includes "counting and thresholding the number of repeated arrivals of identical events." Jakobson, [0188]. A POSITA would have understood that threshold-based events are *conditional* events because they are conditioned on the comparison to a threshold. For instance, a counted and thresholded event would only be generated if (and therefore, is *conditional*) the number of repeated arrivals of the event are identical, and meet the threshold requirement.

131. Thus, creating events by comparing incoming sensed data to a threshold, or inferring new events using Boolean operators and conditional predicate terms discloses or at least renders obvious “*generating conditional events.*”

132. Accordingly, Jakobson’s disclosure of generating high-level inferred events by event correlation discloses *generating conditional events.*

“context aware information”

133. In a **first example**, Jakobson’s situation manager 9, including the case-based reasoning services/engines, generates situations (*context aware information*). See Jakobson, FIG. 7, CBR Services 34. Jakobson describes a situation as “a collection of one or more events that are related by at least one of temporal, spatial, logical, arithmetic, cause-and-effect, or modal constraints.” Jakobson, [0025]. Similarly, Jakobson describes a “situation” as a “time-dependent state of a system” that are described by “domain-specific relations and qualities.” Jakobson, [0025].

134. Jakobson’s situation manager generates or identifies a situation when a collection of incoming events from the event correlation module the requirements of a situation template selected “from a predefined catalog of situation templates.” Jakobson, [0052], [0026]. Accordingly, a situation can have a higher-level modality than the correlated events discussed above. For instance, referring to FIG.

3, Jakobson explains that a situation may correspond to a high-level state, such as “power disruption” or “chemical release.” Jakobson, [0177].

Situation: 4591

Threat: power disruption

Location: Building 12

Date & Time: 2/29/2003 12:01:11

Priority: 4

- high risk individuals sighted near area =>
- traffic level: 30% peak
- power: ok
- crime in vicinity: none reported
- perimeter security: unknown
- power backup: ok

Situation: 5362

Threat: chemical release

Location: Building 1, Building 8

Date & Time: 2/28/2003 11:41:22

Priority: 8

- high risk individuals sighted near area =>
- traffic level: 30% peak
- chemical agent purchases: { X: none, Y: unknown }
- chemical agent thefts: { X: yes => Y: none }
- crime in vicinity: none reported
- chemical handling equipment purchases: none
- medical incidents for chemical agents: unknown
- chemical agent sensors: updating

Jakobson, Fig. 3 (partial annotated)

135. Jakobson’s situations are consistent with the ’232 patent’s description and examples of “context aware information,” which describe a state of a system (e.g., “high probability of problem in the heating machine”) that meets a collection of pre-defined criteria (e.g., average temp. below 14 C, humidity 80%, nighttime, latest problem related to gas leakage and power failure):

Classification	Analysis of sensed data and generation of information	Description
Conditional Events generated by USN middleware apparatus	Provide alarm (“low temperature: 13.5”) for average temperature value	Application has already set a predetermined rule: alarm is generated to application program when average temperature for latest 1 hour is below 14° C.
Context aware information generated by USN middleware apparatus	Provide an imminent report about the given position (“There is high probability of problem in the heating machine; Check gas leakage or breakdown of the electricity”)	USN middleware found the related rule and contents from its internal storage, as following. 1) There is high probability that problem will occur in heating facility when average temperature is below 14° C., current humidity is 80%, and it is night. 2) The latest problem occurring in the heating machine was related to the gas leakage and power supply failure.

’232 patent, Table 1 (partial annotated)

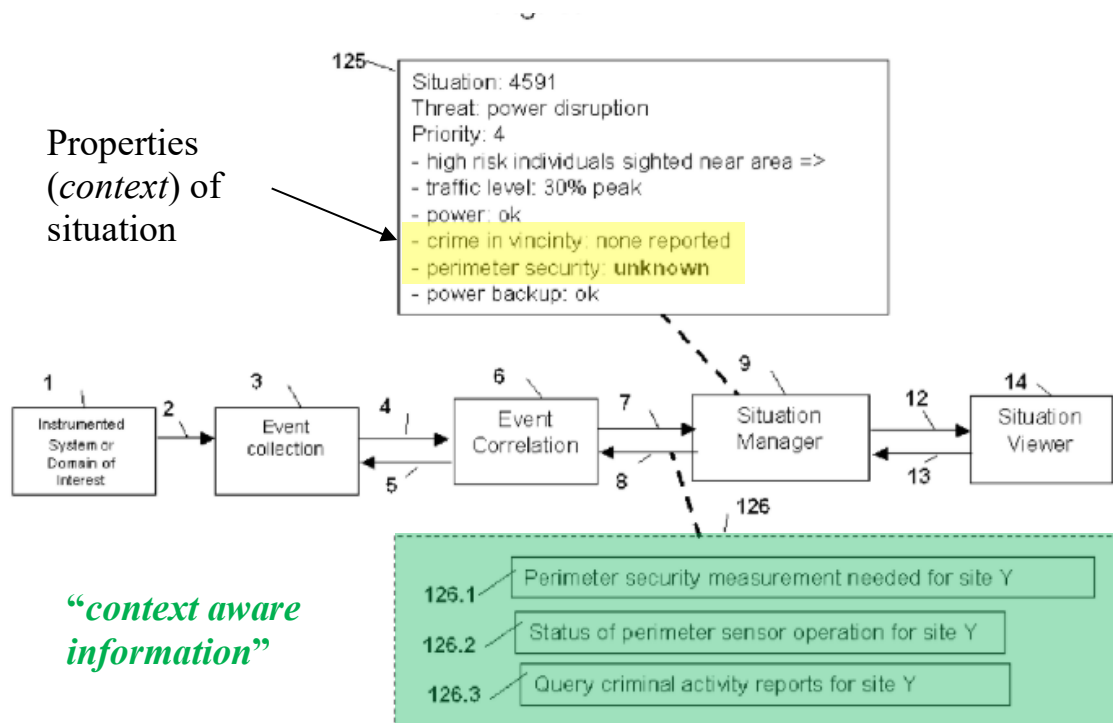
136. The situations are “*context aware information*” because they are based on the context provided by the criteria of the situation template, as in the ’232 patent. ’232 patent, Table 1.

137. In a **second example**, Jakobson discloses that the situation manger also generates the “associated **context**” for a situation. Jakobson, [0027]-[0029]. Jakobson refers to this context as “**context-sensitive information.**” Jakobson, [0282], which the situation manager provides “to the event correlator.” Jakobson, [0282].

138. As an example, of “context-sensitive information,” Jakobson explains that situations are “given a degree of confidence” which may be “less than perfect.” Jakobson, Abstract, [0282]; *see also* [0298], [0322] (using statistical methods to assign confidence level to a situation). Jakobson explains that SBM may “take action” to provide instructions (*context aware information*) to the event correlator (EC) to request information that would increase the confidence level in the situation (e.g., missing parameter values):

The situation may contain parameters which, if provided, may increase the confidence level of the situation. Thus, the SBM may **take action** to be provided the missing information or provide instructions to external resources, such as the EC, to provide such information. Jakobson, [0282].

139. In a more detailed example of requesting information referring to FIG. 18 below, Jakobson describes the situation manager feeding back operational context to the event correlator, where the operational context is related to a “need for additional information based on the context provided by the current situation in affect.” Jakobson, [0299]-[0300]. In this example, the “context” (yellow) includes the properties or parameter values (e.g., perimeter security, crime in vicinity) of the situation. The information 126.1-126.3 are messages or requests that are created and provided to the event correlator 6 based on the properties the situation manager identifies.



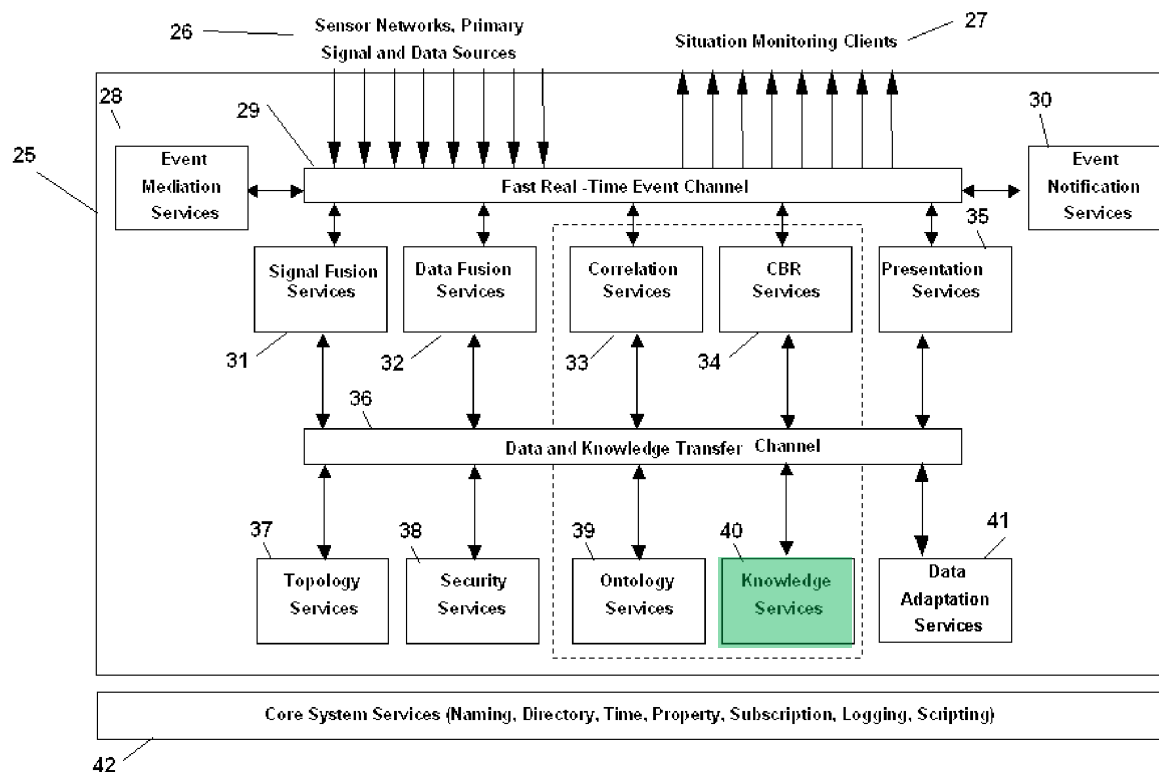
Jakobson, FIG. 18 (annotated)

140. Thus, the information 126.1, 126.2, 126.3 (green) is *context aware information* because it is “**based on the context** provided by the current situation in affect.” Jakobson, [0299]; *see also*, FIGS. 19 and 20 (additional *context-aware information* examples similar to FIG. 18, including withdrawal of query, indication that sensor readings are inconsistent, and indication of inferred impersonation attempt).

141. In a **third example** of “*context aware information*,” Jakobson describes providing ranking values for different situations, depending on what types of parameters are available (e.g., parameter x or parameter y). Jakobson, [0296]. Based on information about which parameters are available, “alternative situations can be ranked with certainty factors based on the available information.” Jakobson, [0297]. For instance, “[t]he situation produced when both x and y are available would have higher rank than a situation produced when only x is available, all else being equal.” Jakobson, [0297]. In this example, the *context* is information about which parameters are available (e.g., x, y, or both). The ranking of the situations based on which parameters are available is therefore *context aware information*.

“knowledge contents”

142. Referring to Fig. 7, Jakobson explains that the SBM services include a “knowledge service 40,” shown in green below. Jakobson, [0217], [0219].



Jakobson, Fig. 7

143. The knowledge service provides “domain modeling, knowledge representation, and reasoning facilities used by the Event Correlation Service 33 and the Case-Based Reasoning Service 34.” Jakobson, [0217].

144. Fig. 8 and the corresponding text of Jakobson provide more details about the function and structure of the knowledge services, including “knowledge discovery agents” and “knowledge bases” (green) that provide “semantic information...including synonyms, antonyms,” and various other “semantic relationships.” Jakobson, [0223].

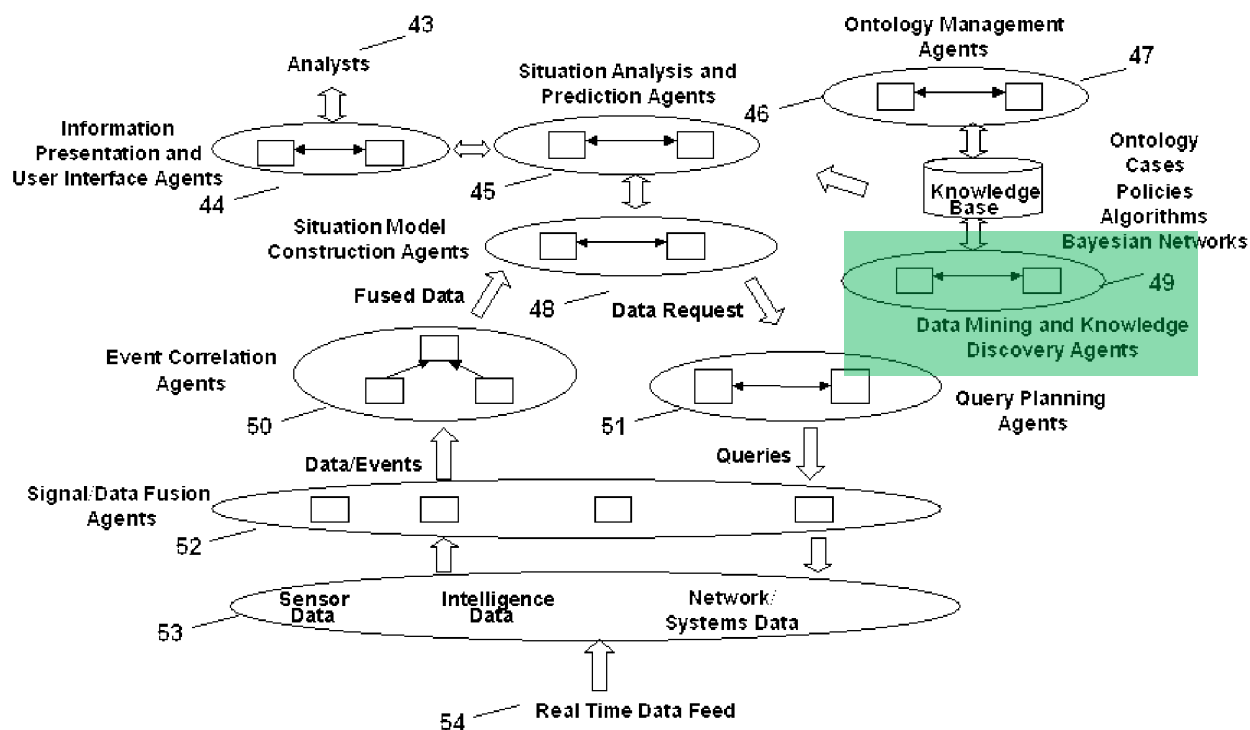


Fig. 8

Jakobson, Fig. 8

145. The information contained in the knowledge services, ontology services, or both, includes *knowledge contents*.

146. Jakobson explains that the knowledge base and ontology services can be modified by the situation viewer 14:

In FIG. 4, the situation viewer 14 may also operate on situation-based management 11 via communications channel 13. These operations include changing properties of situations, creating new situations, **modifying** the knowledge base of situation-based management, and **modifying** the ontology of the situation manager.

Jakobson, [0180].

147. Thus, Jakobson discloses *generating* (by modifying) *knowledge contents* stored in the knowledge base.

148. Although Jakobson is somewhat limited in its discussion of how the knowledge base and ontology is generated, it would have been obvious for the SBM middleware to generate contents of the knowledge base over time, based on the correlated events and situations. A POSITA would have understood that such historical information can be leveraged to improve the performance and accuracy of situation management, and to provide more reliable, useful information. For instance, the knowledge contents may include a “historic sequence of situation transmissions,” or a “history of past known situations” indicating how situations evolved, how many times a situation was detected, and the attending circumstances could be used to more accurately identify future situations, and to predict future situations that could stem therefrom. Jakobson, [0174], [0222]-[0223] (prediction agents identify potential future situations based on knowledge bases) [0224]-[0239] (situation templates include fields regarding the “history of the situation” and “projections or predications of this situation”), [0310] (situation transition information tracks how situations change over time), [0322].

149. As another example shown in FIG. 3, Jakobson’s system generates and provides a set of properties (*knowledge contents*), shown in green, which are associated with a situation, and includes them as part of a situation report:

Situation: 4591

Threat: power disruption

Location: Building 12

Date & Time: 2/29/2003 12:01:11

Priority: 4

- high risk individuals sighted near area =>
- traffic level: 30% peak
- power: ok
- crime in vicinity: none reported
- perimeter security: unknown
- power backup: ok

Situation: 5362

Threat: chemical release

Location: Building 1, Building 8

Date & Time: 2/28/2003 11:41:22

Priority: 8

- high risk individuals sighted near area =>
- traffic level: 30% peak
- chemical agent purchases: { X: none, Y: unknown }
- chemical agent thefts: { X: yes => Y: none }
- crime in vicinity: none reported
- chemical handling equipment purchases: none
- medical incidents for chemical agents: unknown
- chemical agent sensors: updating

Jakobson, Fig. 3 (partial annotated)

150. As yet another example of *knowledge contents*, Jakobson discloses generating “situation transition graphs” to demonstrate how situations evolve over time:

According to another aspect of the invention, a method and apparatus are provided for **creating** and utilizing **a situation transition graph (STG)** in situation-based management... A situation transition represents a sequential, iterative or serial progression from one situation to another situation.

Jakobson, [0309]-[0310].

151. Accordingly, in addition to the “knowledge bases” described above, Jakobson’s situation properties, and situation transition graphs are *knowledge contents*.

152. Thus, Jakobson discloses or renders obvious this limitation.

[1.5] *providing the information service to an application program, the application program providing a user with information related to environment where the sensor network is installed.*

153. Jakobson discloses providing a “situation viewer” with situation reports, events, and other situation management information (*the information service*) to a user interface application:

...situations, situation reports, extracts of situations, abstracts or summaries of situations, or descriptions of situations through communications channel 12 such as **a user interface, an instant message, an email, and a voice, image, or video input system.**

Jakobson, [0178], [0179] (user interface presents “situations, situation reports, extracts of situations, descriptions of situations, and events”).

154. Accordingly, the information service provided by the situation view provides situation reports, extracts of situations, and other situation-related information. FIG. 14 illustrates the situation viewer (orange), which can include an information system 14.2 and a user interface 14.3.

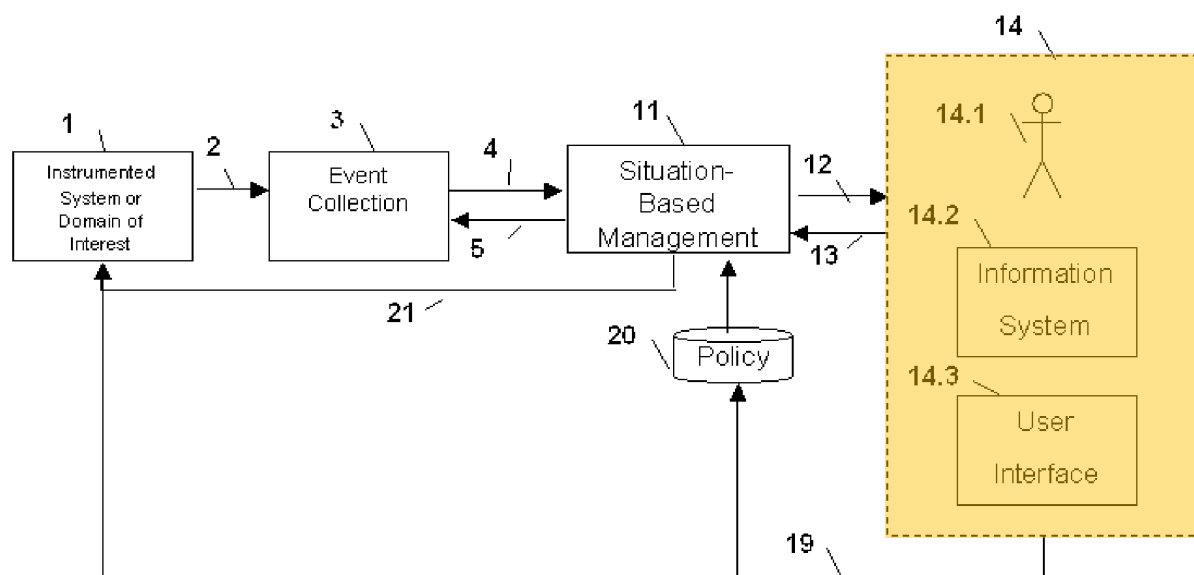


Fig. 4

Jakobson, FIG. 4 (annotated)

155. Jakobson explains that this *information* (e.g., situation reports) is viewable via a user interface (14.3), and that the user can interact with and manipulate the information:

In FIG. 4, the user interface 14.3 provides at least one modality by which **situations, situation reports, extracts of situations,**

descriptions of situations, and events are presented. User interface modalities include visual, textual, tactile, and auditory. There may be various means to **control the presentation such as a keyboard, a pointing device, or voice input.**

Jakobson, [0179].

156. Similarly, the “information system 14.2” (also *an application program*) may receive “a situation, a situation report, an extract of a situation, an abstract or summary of a situation or a description of a situation from situation-based management 11.” Jakobson, [0178]. It would have been obvious for the situation viewer (or viewers) to include both an information system, and a user interface. For instance, Jakobson explains that a purpose of the situation manager is to “provide[] information to human **users and external systems.**” Jakobson, [0026]. It would have been obvious, and expected, for the situation reports, events, and other information output by the SMB services to be provided both to a user interface *application program*, and to an external information service *application program*. These services for collecting and viewing situation information can provide different functionalities and suit different purposes, and a POSITA would have been motivated to provide these functionalities to the clients of the system.

157. To the extent Jakobson does not explicitly or implicitly disclose that the information system, user interface, or communications/multimedia programs (e.g., instant message, email) include *an application program*, this would have

been obvious. It was well-understood that user interfaces and communications applications and services included software applications to provide an interface (e.g., a graphical user interface) for a user to view, manipulate, and request information. In that regard, Jakobson explains that aspects of involve human input and well-known “software and communications technology”:

In addition, a communications channel may be performed or aided by human intervention, where **a human operator takes output from the SM** and provides the output as input to the EC. **These and other forms of communication will be familiar to those versed in the practice of software and communications technology.**

Jakobson, [0074]; *see also*, [0004] (explaining that managing domains of sensors is “typically [] implemented in software on networked data processing systems”), [0215] (SBM constructed as distributed services software), [0220] (SBM constructed as agent-based software).

158. Finally, because this situation-related information is based on the sensor data (events) provided by the sensors of the domain, the information is *related to environment where the sensor network is installed*.

159. Accordingly, providing situation reports or other situation-related information to the user to an information system, a user interface, and/or through communication applications (e.g., instant message, email) discloses or renders obvious “*providing the information service to an application program, the*

application program providing a user with information related to environment where the sensor network is installed.”

b) Claim 2

[2.0] *The system as recited in claim 1, wherein the ubiquitous sensor network middleware comprises: a sensor network abstraction unit for extracting the sensed data from the message received from the sensor network data transmitter;*

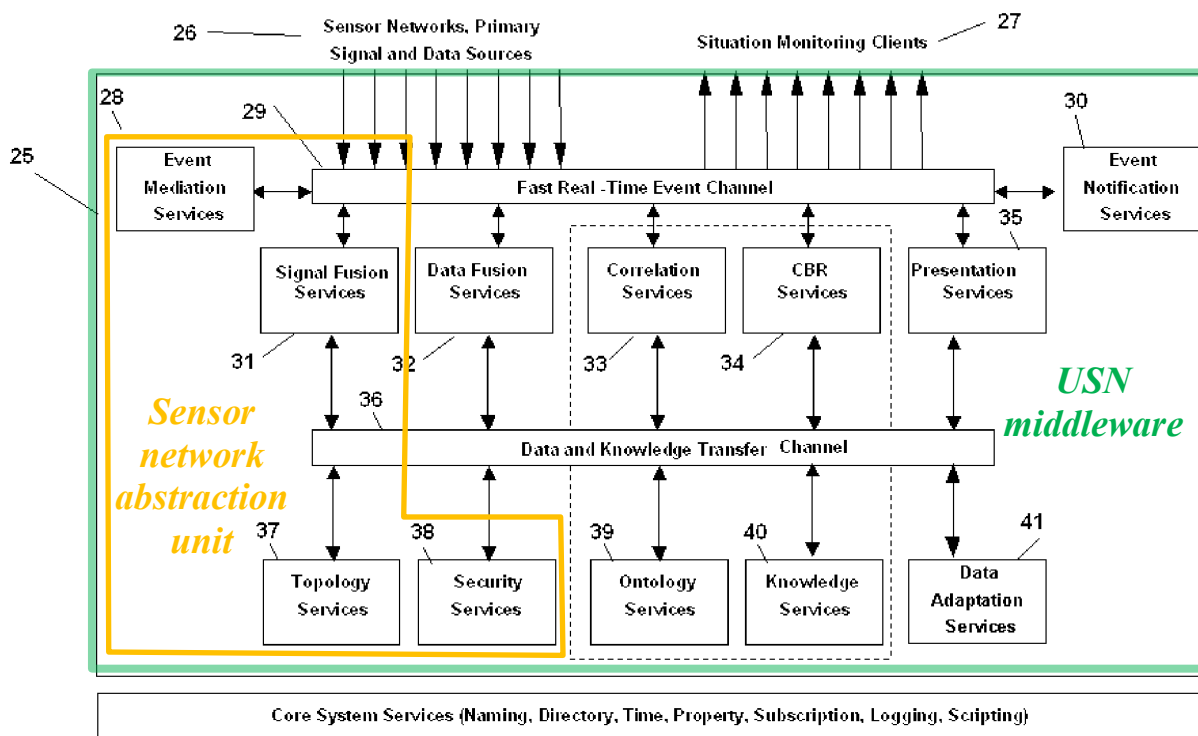
160. As explained above at [1.2], Jakobson and Tsetsos disclose or render obvious the *sensed data* being communicated *in message form*, and *from the sensor network data transmitter*. As also explained above, Jakobson refers to sensed data as “events,” which include digital signal data. Jakobson, [0012].

161. Jakobson’s SBM system includes a “signal fusion” module that “extract[s] information” from the signals (*sensed data*) provided by the sensors. Jakobson, [0010]. For instance, the objects of signal fusion includes “identification of one or more targets” or target locations, “tracking one or more targets,” and to “convert and **extract** low-level signals into **high-level data** more suitable for processing.” Jakobson, [0010]. Thus, the signal fusion module *extract[s] the sensed data from the message*.

162. Referring to the distributed system architecture of the situation manager shown in FIG. 7 below, Jakobson explains that the fast real-time event channel 29 in conjunction with the event mediation module 28 use communication protocols to provide the sensor data to the signal and data fusion modules:

There are four real-time application services: Signal Fusion 31, Data Fusion 32, Event Correlation 33, and CBR 34. These services are connected to **a fast real-time event channel 29**. **The event channel 29 transports events between services**, in which the transport is reliable, fast, and secure. Event Mediation 28 performs **the connectivity and protocol conversion functions** so that sensor and intelligence data can reach **the Signal Fusion 31, Data Fusion 32, and Event Correlation 33 services**.

Jakobson, [0216].



Jakobson, FIG. 7 (annotated)

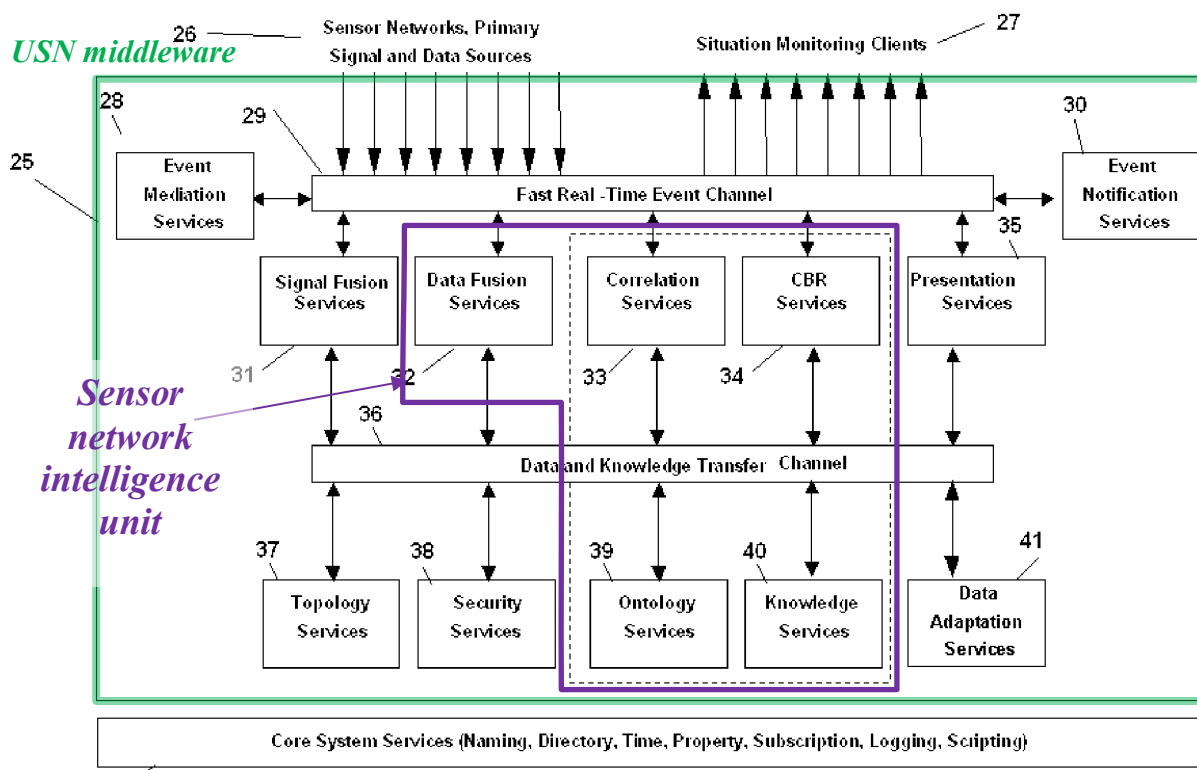
163. Accordingly, event channel 29 and event mediation 28 services provide an interface between the sensor networks and the signal and data fusion modules, which extract the sensed data from the events received from the sensors.

164. Thus, Jakobson's signal and data fusion modules, event mediation module 28, and the event channel 29, disclose or render obvious the claimed *sensor network abstraction unit for extracting the sensed data from the message received from the sensor network data transmitter.*

[2.1] *a sensor network intelligence unit for extracting the object and environment information by analyzing the sensed data extracted by the sensor network abstraction unit, and generating the conditional events including notification and alarm about a certain condition, and the context aware information from the extracted object and environment information; and*

165. As explained above for limitation [1.3], the data fusion module and/or the event correlation module extract the “*object and environment information*” by “*cleaning, classifying, and integrating the message.*”

166. As also explained above for limitation [1.4] the event correlation module and the situation manager (CBR Services) generate the *conditional events* and the *context aware information*.



Jakobson, FIG. 7 (annotated)

167. Jakobson explains that the data fusion and event correlation services perform operations based on the data provided by signal fusion services (*by analyzing the sensed data extracted by the sensor network abstraction unit*). Jakobson, [0013] (events from signal fusion are provided to data fusion and then to event correlation).

168. Thus, Jakobson's data fusion services, correlation services, and situation management (CBR services) modules disclose or render obvious the claimed *sensor network intelligence unit*.

"extracting the object and environment information"

169. Jakobson explains that the event correlation module 6 “**extracts** the essential **properties** of the events” (*object and environment information*), provided from the sensor networks via the event collection module 3. Jakobson, [0175].

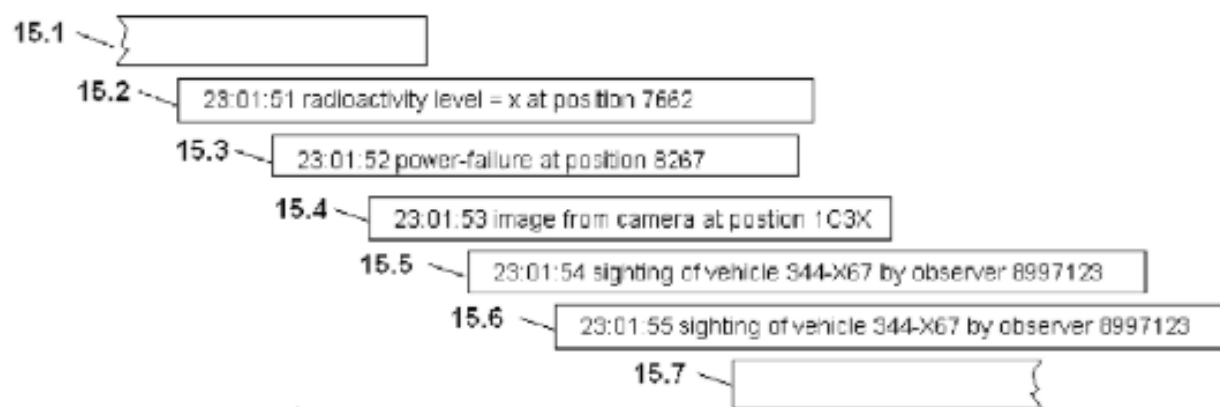
170. The extracted “properties” include *object and environment information* because the properties represent a condition of the domain and are based on the sensed data from the domain. Jakobson, [0170]-[0173], FIG. 3. Jakobson further explains that “properties” extracted by the event correlation module “may contain actual values, values with likelihood estimates...calculated values....” Jakobson, [0177]. Thus, Jakobson discloses that the data fusion services and/or the event correlation services (corresponding to *the sensor network intelligence unit*) *extract[] the object and environment information.*

“generating the conditional events including notification and alarm about a certain condition”

171. **First**, as explained above at limitation [1.4], the event correlation generating high-level inferred events, the situation manager creating situations based on a comparison of events to one or more situation templates, or both, disclose “*generating the conditional events.*” Further, as also explained above at limitation [1.4], Jakobson’s situation manager (part of *the sensor network intelligence unit*) generates “*the context aware information from the extracted object and environment information.*”

172. Second, Jakobson explains that both the correlated high-level events (*conditional events*) include a “*notification and alarm about a certain condition.*” For instance, regarding event correlation, Jakobson’s events are referred to as “notification[s],” and may include “human-readable text.” Jakobson, [0012].

173. In one example, events may indicate higher level alerts of a condition, such as “it is suspect that sensor 123 exceeded a threshold” (*notification and alarm about a certain condition*). Jakobson, [0176]. FIG. 2 below includes several events 15.1-15.7, each including *a notification about* some state of interest, such as “power-failure at position 8267,” “radioactivity level = x,” and “sighting of vehicle by observer [x].”



Jakobson, FIG. 2 (partial)

174. In a second example, Jakobson explains that a “UNIT-SUPPORT-CORRELATION-RULE” running at the event correlation module results in a message being sent to the situation manager indicating that contact has been lost

(“UNIT_CONTACT_LOST_SITUATION”) with some asset (e.g., a tank).

Jakobson, [0191].

175. A POSITA would have understood that each of these event messages produced by the event correlation 6 includes a “*notification and alarm about a certain condition.*” In that regard, the events that include notification about power failure, radioactivity level, or lost contact are consistent with the ’232 patent’s disclosure regarding “conditional events,” as including a “notification of a specific error and alarm of crucial state,” such as “low temperature: 13.5.” ’232 patent, 5:54-55, 6:13-15. In other words, the ’232 patent describes the “*notification and alarm about a certain condition*” as the content of a single conditional event having a crucial state, rather than two separate events or separate components of an event.

Classification	Analysis of sensed data and generation of information	Description
Conditional Events generated by USN middleware apparatus	Provide alarm ("low temperature: 13.5") for average temperature value	Application has already set a predetermined rule: alarm is generated to application program when average temperature for latest 1 hour is below 14° C.
Context aware information generated by USN middleware apparatus	Provide an imminent report about the given position ("There is high probability of problem in the heating machine; Check gas leakage or breakdown of the electricity")	USN middleware found the related rule and contents from its internal storage, as following. 1) There is high probability that problem will occur in heating facility when average temperature is below 14° C., current humidity is 80%, and it is night. 2) The latest problem occurring in the heating machine was related to the gas leakage and power supply failure.

'232 patent, Table 1 (partial annotated)

176. Thus, like the '232 patent, Jakobson's generation of high-level inferred events indicating a low temperature, radioactivity level, or lost contact discloses or renders obvious "*generating the conditional events including notification and alarm about a certain condition.*"

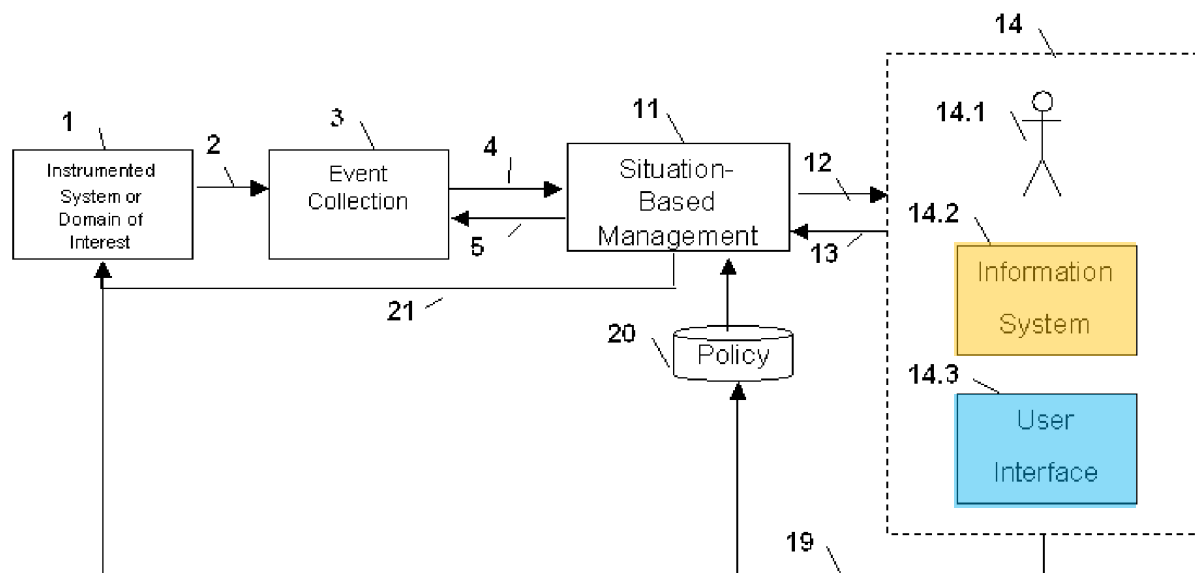
177. As explained above at limitation [1.4], Jakobson's situation manager (part of *the sensor network intelligence unit*) generates situations (*the context aware information*) based on the data extracted and analyzed by the data fusion

module and event correlation services (*from the extracted object and environment information*).

178. Thus, Jakobson discloses this limitation or at least renders this limitation obvious.

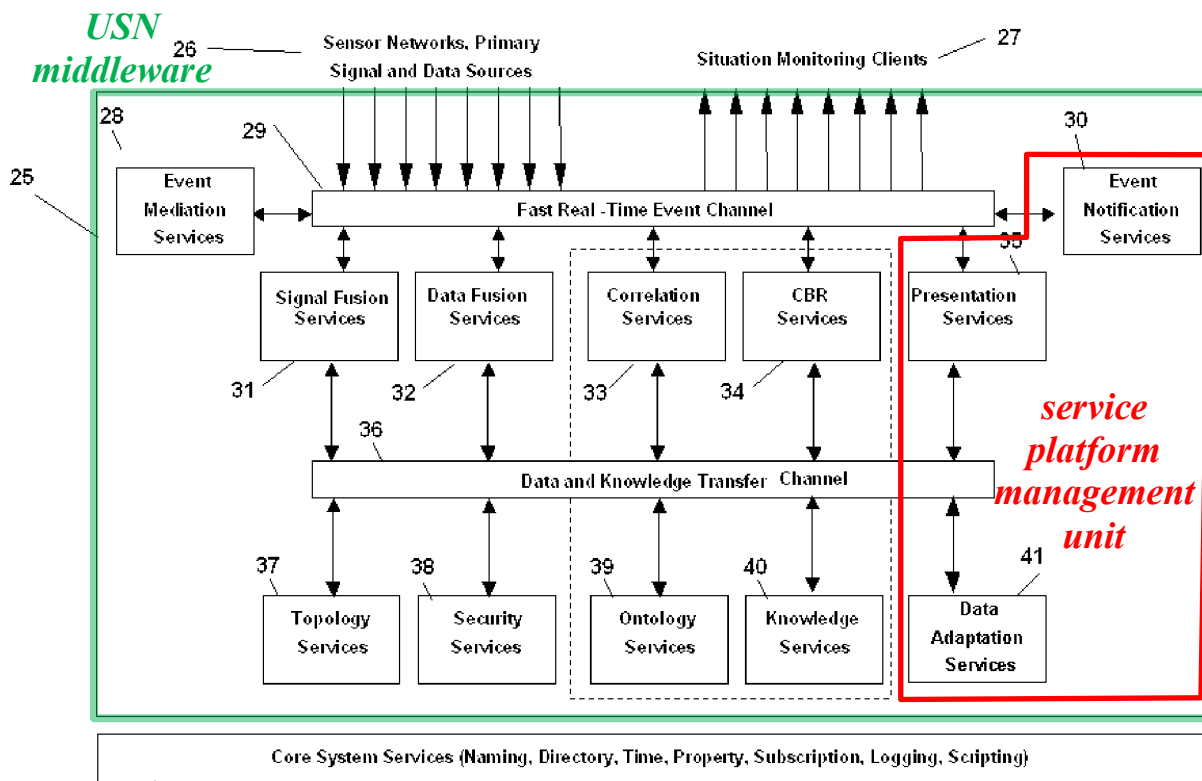
[2.2] a service platform management unit for providing the events and the context aware information to the application program in a form suitable for the application program.

179. Referring to FIG. 4, Jakobson explains that the situation viewer 14 includes an “information system 14.2” (*application program*) (orange) that receives situation-related information that has been “converted to a different format” (*in a form suitable for the application program*). Jakobson, [0178], [0179]. Jakobson also explains that situation-related information can be provided to a “user interface” (also *application program*) (blue). Jakobson, [0178], [0179].



Jakobson, FIG. 4 (annotated)

180. Additionally, referring to FIG. 7, Jakobson's middleware includes "Event Notification 30" which "provides facilitates for subscribing to and filtering events." Jakobson, [0216]. The middleware 25 also includes presentation services 35 to "support the user interface" and to present situation to "situation monitoring clients 27." Jakobson, [0218], [0219].



Jakobson, FIG. 7 (annotated)

181. Further, Jakobson explains that the situation viewer is provided with not only situation reports (*context aware information*), but also "events" for presentation to the viewer:

In FIG. 4, the user interface 14.3 provides at least one modality by which situations, **situation reports**, extracts of situations, descriptions of situations, **and events** are presented.

Jakobson, [0179].

182. In the above, the information service is provided to “situation monitoring clients.” A POSITA would have found it obvious for the situation monitoring clients to include software modules facilitating the user interface and information service functionality of the situation viewer 14. In another example, it would have been obvious to implement that functionality in the presentation services or notification services. In either case, deploying the functionality of formatting, filtering, and presenting implementation in an external device that communicates with the middleware services 25 or within the middleware services 25 themselves is an implementation detail, and both options would have been within the skill set of a POSITA. A POSITA would have recognized that implementing the situation report formatting in the middleware services 25 would have reduced the processing and computational demands at the client device, and potentially reducing the amount of information that needs to be transmitted over a data network to the client’s device or system.

183. Jakobson’s event notification services 30 and presentation services 35 are therefore responsible for formatting, filtering, and presenting situation information (e.g., reports, abstracts) in a way that is compatible with the

application running at the situation viewer (user interface 14.3 and/or information system 14.2), and useful to help a viewer understand and respond to situations.

184. Thus, Jakobson's presentation services 35 and event notification services 30 disclose or render obvious the *service platform management unit for providing the events and the context aware information to the application program in a form suitable for the application program.*

c) Claim 3

[3.0] *The system as recited in claim 2, wherein the sensor network abstraction unit comprises: a sensor network interface for extracting the sensed data by referring to a message transfer protocol;*

185. First, as explained above for limitation [2.0], the real-time channel 29 receives sensor data from sensor networks using the event mediation 28 to perform protocol conversion functions.

186. Second, as explained above for limitation [1.2], Jakobson and Tsetsos render obvious *sensor network data transmitter* (a gateway) transmitting messages to the SBM middleware in packet form over the internet using TCP/IP (*a message transfer protocol*). A POSITA would have understood that the receiving device *refer[s]* to the TCP/IP to obtain the data or payload of the packets (for extracting the sensed data). Newton, 812-813 ("TCP/IP" transmits "messages" as "packets"), 493 ("Internet Protocol (IP)...routes outgoing **messages**, and recognizes incoming **messages**"), 611-612 (when packets arrive at destination, their headers are

detached and discarded to obtain the payload); Tanenbaum, 41-44 (TCP/IP packet handling).

187. TCP/IP is the standard protocol for receiving data packets over the Internet, and a processing system that intends to receive the data packets for processing would have been expected to have the communication and processing capabilities to extract the payload from an IP packet in accordance with the IP protocol (*a message transfer protocol*). Similarly, even for local area networks (LANs) that do not necessarily need to transmit data over the broader Internet (e.g., across great distances), TCP/IP protocols may still be used, in addition to other communication protocols (e.g., Wi-Fi (IEEE 802.11) that have their own message transfer protocols that can exist on top of, or alongside, TCP/IP.

188. It would have therefore been obvious for the real-time event channel 29 and/or the event mediation 28 (*a sensor network interface*) to refer to the TCP/IP (or a similar standard networking protocol) to remove headers from sensor network packets and obtain the data payload (*for extracting the sensed data by referring to a message transfer protocol*).

189. Additionally, Jakobson explains that the situation-based management methods involve “**a protocol** for providing events from the [event collector] to the [situation manager].” Jakobson, [0056], [0057]. Jakobson explains that distributed

system architectures, for instance, use “communication protocols” (*message transfer protocol[s]*) to transfer information between system components:

The practice of software architecture for distributed systems includes using standard middleware services or components with well-defined functionality and **standard inter-component communication protocols**.

Jakobson, [0215].

190. For instance, Jakobson discloses using communication frameworks like “CORBA (Common Object Request Broker Architecture), Jini, Web Services, J2EE (Java 2 Enterprise Edition), JAIN SLEE (Service Logic Execution Environment) and DCOM (Distributed Component Object Model).” Jakobson, [0215]. These frameworks work with communication and application protocols, and are used as the “infrastructure of distributed systems” including those that provide Middleware. Jakobson, [0215].

191. The communication protocols are used by Jakobsons’s “event mediation 28,” which “performs the connectivity and **communication protocol conversion functions** so that sensor and intelligence data can reach” the fusion and event correlation services. Jakobson, [0216]. Jakobson’s event channel 29 relies on the event mediation’s protocol conversions (*referring to a message transfer protocol*) to provide the sensed data to the signal fusion services for extraction (*for extracting the sensed data*).

192. Accordingly, the event channel 29, the event mediation service 28, or both, disclose or render obvious the claimed *sensor network interface* for *extracting the sensed data by referring to a message transfer protocol*.

193. Additionally, it would have been obvious for the event mediation module 28 to include, or function as, a *sensor network interface* in view of Tsetsos. Tsetsos describes techniques for communicating with and managing wireless sensor networks (WSNs) using application program interfaces (APIs) and standard communication protocols for Sensor-Based Services (SBS). Tsetsos, 1-3. Tsetsos describes using communication protocols like “Parlay/OSA (Open Services Access)” and “OSGi (Open Services Gateway Initiative)” to handle communications between sensor devices and an application platform. Tsetsos, 3, 5. APIs based on those communication protocols can facilitate “the interaction between the applications and the network elements,” including “sensor data retrieval.” Tsetsos, 3.

194. Tsetsos further explains that the APIs specified by standards like Parlay/OSA or OSGi may be used for “middleware” to provide “unified handling of [WSNs].” Tsetsos, 5. These APIs provide developers with functionalities for “user interaction, mobility management, user terminal capabilities, location, etc.... without compromising the overall network security.” Tsetsos, 5.

195. It would have been obvious for the event mediation 28 (*sensor network interface*) in Jakobson to rely on APIs and standard protocols like those described in Tsetsos (e.g., Parlay/OSA, OSGi) to provide a comprehensive sensor network handling framework to retrieve (*extract*) sensor data. *See supra*, § V.C.

196. Additionally, as similarly explained above with respect to limitation [2.0], the signal fusion services include “**extracting** information from the signals” (*extracting the sensed data*) received from the sensor networks. Jakobson, [0010].

197. Because Jakobson also discloses that its middleware uses “standard inter-component communication protocols” and “protocol conversion functions” to provide and extract information from the sensor networks, Jakobson discloses or at least renders obvious the signal fusion services extract the sensor data *by referring to a message transfer protocol*. Jakobson, [0215], [0216].

198. Thus, the signal fusion services also render obvious the claimed *sensor network interface for extracting the sensed data by referring to a message transfer protocol* for similar reasons as explained above.

199. Further, Tsetsos also describes communicating sensor data from wireless sensor networks to middleware services over the internet:

The gateway node has more processing, energy and communication capabilities than the other nodes and is the connecting link between the sensor network and the WAN (i.e., mobile network or **Internet**). Hence, this node should be capable of communicating with the mobile

network infrastructure, either directly (GSM/GPRS modem) or indirectly (Internet modem and Gateway GPRS Support Node - GGSN).

Tsetsos, 3.

200. A POSITA would have understood that these standardized, layered protocols (e.g., IP for internet communications) involve *extracting* data from packets (e.g., removing headers from IP packets or other high-layer packets to obtain payload).

201. Further, even for intra-system communications that may not travel over the internet, Jakobson discloses using communication protocols to provide events and information from one module to another. For instance, Jakobson explains that the components of its system are “encapsulated” as software modules using software languages and models:

This type of distributed software architecture allows the building of open, scalable, and customizable systems. The encapsulation of the idiosyncrasies of components and the addition, replication, and replacement of components provide an effective environment for developing multi-paradigm, fault-tolerant, and high-performance systems. Various software technologies can be used as the infrastructure of distributed systems, including CORBA (Common Object Request Broker Architecture), Jini, Web Services, J2EE (Java 2 Enterprise Edition), JAIN SLEE (Service Logic Execution Environment) and DCOM (Distributed Component Object Model).

Jakobson, [0215].

202. A POSITA would have understood that these intra-system protocols also involve *extracting* data from messages or packets.

203. Referring to the distributed system architecture of the situation manager shown in FIG. 7, Jakobson explains that the fast real-time event channel 29 relies on protocol conversion (*referring to a message transfer protocol*) provided by the event mediation module 28:

There are four real-time application services: Signal Fusion 31, Data Fusion 32, Event Correlation 33, and CBR 34. These services are connected to **a fast real-time event channel 29. The event channel 29 transports events between services**, in which the transport is reliable, fast, and secure. Event Mediation 28 performs **the connectivity and protocol conversion functions** so that sensor and intelligence data can reach **the Signal Fusion 31, Data Fusion 32, and Event Correlation 33 services**.

Jakobson, [0216].

204. Thus, Jakobson's event channel 29 relies on the event mediation's protocol conversions (*referring to a message transfer protocol*) to provide the sensed data to the signal fusion services for extraction (*for extracting the sensed data*).

205. Accordingly, the event channel 29, the event mediation service 28, or both, disclose or render obvious the claimed *sensor network interface for extracting the sensed data by referring to a message transfer protocol*.

206. Additionally, as similarly explained above with respect to limitation [2.0], the signal fusion services extract sensed data from the events received from the sensor networks:

Processed or unprocessed signals may be provided to other equipment which performs **signal fusion operations such as extracting information from the signals**, filtering the signals, converting the signals to digital format, eliminating redundant information, and combining multiple signals including historical signals into visual or audio patterns. One object of signal fusion in situation-based management is identification of one or more targets, identification of target location, speed and directions, and tracking one or more targets. Another object of signal fusion in situation-based management is to distribute specialized signal processing functions for efficient transmission. Another **object of signal fusion** in situation-based management is to **convert and extract low-level signals into high-level data more suitable for processing**.

Jakobson, [0010].

207. Because Jakobson also discloses that intra-component communications (e.g., from sensor networks to event collection to event correlation, etc.) rely on standardized protocols to provide and extract information,

Jakobson discloses or at least renders obvious, alone or in view of Tsetsos's standardized APIs and associated protocols, the signal fusion services extract the sensor data *by referring to a message transfer protocol*.

208. Thus, alternatively or additionally to the event mediation 28 and/or channel 29, the signal fusion services render obvious the claimed *sensor network interface for extracting the sensed data by referring to a message transfer protocol* for similar reasons as explained above.

[3.1] *a sensor network interface protocol storage for managing a rule for the message transfer protocol; and*

209. As explained above for limitation [3.0], Jakobson's event mediation module 28 uses "connectivity and communication protocol conversion functions" to transfer data to the signal and data fusion services. Jakobson, [0216]. The "connectivity and communication protocol conversion functions" performed by the event mediation module 28 disclose or render obvious a *rule for the message transfer protocols*.

210. For instance, it would have been obvious for the event mediation module 28 to include, or have access to, *storage* that stores instructions on how to extract data from IP packets, and manage the sensor networks using the APIs (e.g., Parlay/OSA). For heterogeneous networks that include a variety of different types of equipment and data processing systems, a POSITA would have understood that the data collection modules would need to include, or have access to, *storage*

including rules for identifying the different types of data for different protocols, and how to request and retrieve the payload for each type. *See* Jakobson, [0007] (collecting and combining data from a variety of sensors using a variety of data processing networks); Tsetsos, 5, 6 (challenges of heterogeneous wireless sensor networks); *supra*, § V.C (describing role of communications protocols (e.g., TCP/IP) and APIs in facilitating communication and interactions in diverse networks). Indeed, Tsetsos explains that one of the very purposes of “middleware” is to facilitate unified handling of information from heterogeneous sources (e.g., sensors). Tsetsos, 5, 8.

211. Thus, it would have been obvious for Jakobson’s event mediation 28, which is responsible for protocol conversion functions, to include or be connected to *storage* that stores and provides access to (*manages*) *a rule for the message transfer protocol*.

212. Additionally, Jakobson explains that the data fusion services convert data to a uniform format:

Events from sensors and signal fusion equipment are provided to other data processing equipment. This data processing equipment may perform data fusion operations on the events such as filtering, translating, cleansing, aliasing, aggregating, and **converting to a uniform representation**.

Jakobson, [0013].

213. Because the data from the data fusion services is provided to event correlation and situation management modules, a POSITA would have understood that “uniform representation” refers to the format and structure of the sensor data for communication and processing purposes, and not to the more user-facing aesthetic “representation” that happens at the application program level.

214. It would have been obvious for the data fusion services to include *storage* to store communication protocol rules that facilitate the converting of the data to the uniform representation. Thus, the data fusion services also render obvious the claimed “*sensor network interface protocol storage for managing a rule for the message transfer protocol*” for similar reasons as explained above.

[3.2] a sensor network monitor for monitoring a state of the sensor network, based on the data received from the sensor network interface.

215. Jakobson explains that managing a domain, including a collection of sensors, involves “monitor[ing]” the domain:

A domain may be instrumented with sensors, probes, surveillance equipment, monitoring apparatus, management apparatus, and other means by which to monitor and report the characteristics of a domain. Hereafter, the term “sensors” is used to refer to one or more of these means of monitoring a domain. Those familiar with the practice of **managing a specific domain** will be able to identify other means and apparatus by which **to monitor and report the characteristics of the domain**.

Jakobson, [0009].

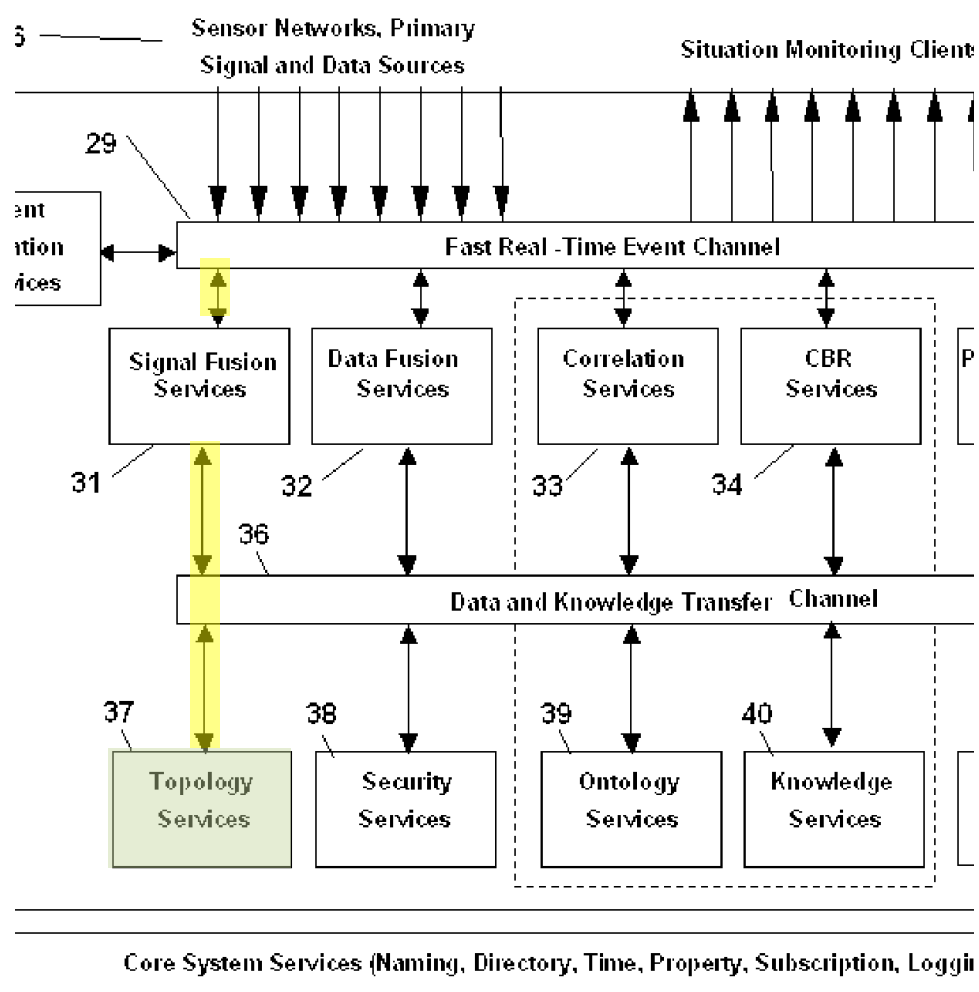
216. Monitoring for data from the sensor networks includes monitoring a state of the sensor network. In that regard, the receipt of data from the sensor networks indicates that the network is in a state of being connected with, and providing data to, the event collection module and SBM system.

217. Further, Jakobson discloses more specific components that collect different kinds of information regarding the connections and status of the sensors of the domain. Jakobson discloses Topology services that receives and manipulates information about connectivity and relationships of the system:

Other services available for application use include (i) Topology 37 to store, represent, and manipulate information about connectivity, containment, and structural relationships...

Jakobson, [0218].

218. As shown in FIG. 7, the topology module 37 receives information from the event channel 29 (*sensor network interface*) through the data and knowledge transfer channel 36 and the signal fusion services.



Jakobson, FIG. 7 (annotated)

219. Jakobson explains that topology information and processes can be used in fault management of networks to monitor and identify failures:

In fault management of telecommunications networks, information is continuously collected from network elements and is combined with data processing models of the equipment and the **topology of the network to identify failures to operations technicians**. The information may be pre-processed using event correlation before

combining it with data processing models of the network equipment and network topology.

Jakobson, [0005].

220. Similarly, Jakobson describes monitoring sensor data and information about the topology of a domain:

In autonomic systems, information is continuously collected from sensors and is combined with data processing models and the topology of a domain.

Jakobson, [0008].

221. Jakobson also describes determining domain-specific topologies for situation management:

a method of SBM comprises:...**determining and maintaining topological**, spatial, temporal, causal and other **domain-specific relations** between created situations;

Jakobson, [0033]-[0035].

222. Thus, Jakobson discloses continuously collecting, from the sensors of the domain (*monitoring a state of the network*), topological information about the sensors.

223. To the extent Jakobson does not explicitly disclose that the topological information is collected by the Topology module 37 *based on data received from the sensor network interface*, this would have been obvious. In that regard, as explained in limitation [3.0], the event channel 29, the event mediation

service 28, or both, disclose or render obvious the claimed *sensor network interface*. FIG. 7 shows that the sensor data from the sensor networks is provided to the Topology module 37 through the event channel 29. It would have been obvious for the Topology module to rely on sensor data received and transmitted by the event channel 29 to collect the topology information (e.g., message headers, sensor metadata, message routing information, hopcount, sensor status, etc.). This would have been an expected and efficient way to collect topology information, because it would not require dedicated or redundant communications between the sensor networks and the topology services.

224. Additionally, it would have been obvious for *the sensor network data transmitter* (e.g., Tsetsos's sensor gateway) to collect information about *the state of the sensor network* (e.g., metadata, message headers) from the sensors and to provide topological status reports to the topology module. For instance, Tsetsos explains that wireless sensor networks are “highly flexible...and can be easily adapted to the considered problem. Tsetsos, 2. Tsetsos explains that WSNs adapt to different problems by “**topological** change.” Tsetsos, 2. Tsetsos explains that the gateway uses a standardized gateway communications protocol like OSGi to aggregate sensor data and its accompanying metadata, like device ID, type, and location. Tsetsos, 7. Listing 1.

```
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance" xsi:noNamespaceSchemaLocation="  
USLRequest.xsd" ID="123456">  
  <Query>  
    <RequestedInfo location="false" time-instance="false"  
      device="true" sensor="true" time-duration="false">  
      <SensorList>  
        <SensorType function="NoFunction"  
          type="temperature"/>  
      </SensorList>  
    </RequestedInfo>  
    <QueryFilter>  
      <Device ID="543"/>  
    </QueryFilter>  
  </Query>  
</Request>
```

Tsetsos, Listing 1, p. 7 (annotated)

225. It would have been obvious for the gateway to provide this topological information for the SBM to stay up to date on the status and available resources in the sensor network, especially for WSNs, which are subject to frequent change and adaptation.

226. Furthermore, Tsetsos teaches that mediation layer frameworks like Parlay/OSA and OSGi can facilitate more seamless integration and control over heterogenous sensor networks:

The heterogeneity of the WSN technologies prevents their seamless use by ubiquitous services. We believe that, until the proper standards are established, a mediation layer (middleware) providing unified handling of these networks should bridge the gap. In the case of telecommunication services, a good candidate for this mediation layer

would be a combination of Parlay/OSA and OSGi (Open Services Gateway Initiative)...The APIs specified by Parlay/OSA expose (almost) all the network functionality (user interaction, mobility management, user terminal capabilities, location, etc.) to application developers without compromising the overall network security and exposing the technical peculiarities of the network to service providers.

Tsetsos, 9.

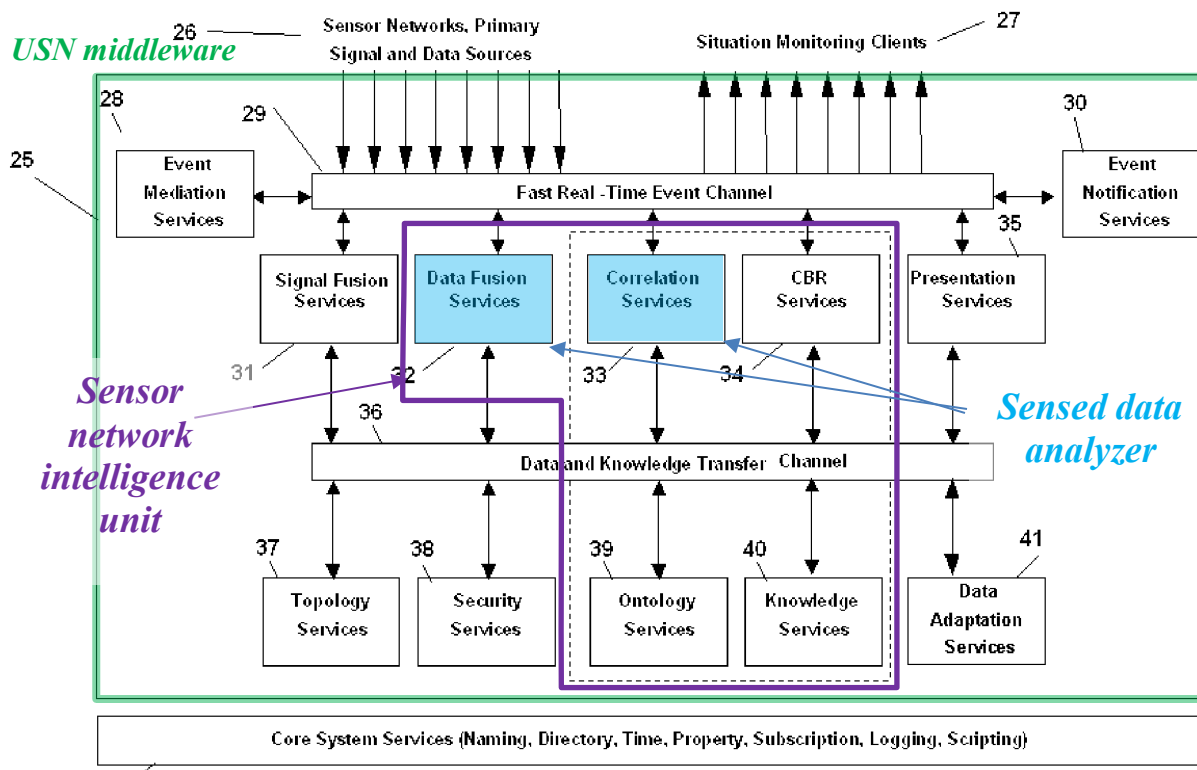
227. It would have been obvious to harmonize the collection of topological information through the gateway in a similar manner.

228. Thus, Jakobson, alone or in combination with Tsetsos, discloses or renders obvious limitation [3.2].

d) Claim 4

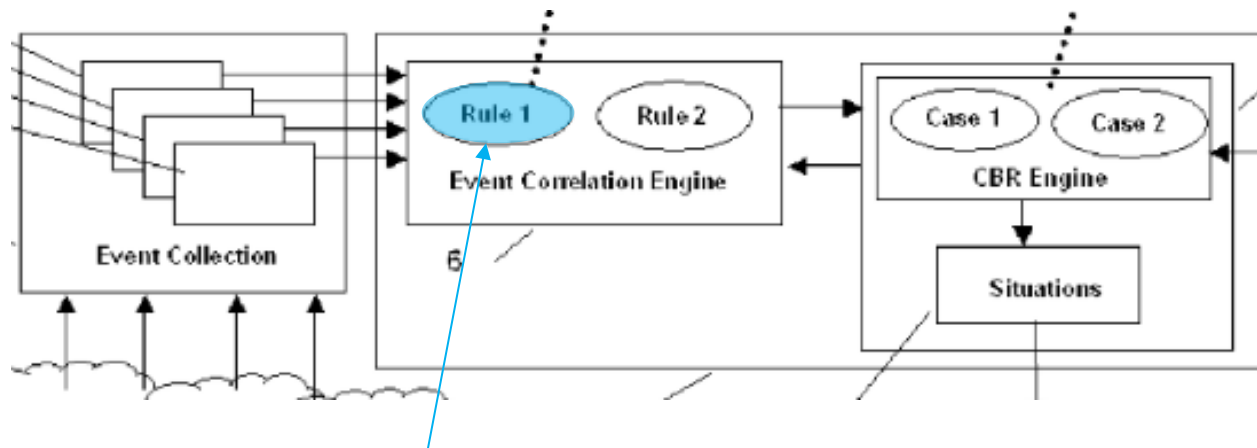
[4.0] *The system as recited in claim 3, wherein the sensor network intelligence unit comprises: a sensed data analyzer for performing an analysis operation to clean, classify and integrate the sensed data processed by the sensor network abstraction unit;*

229. As explained above for limitations [1.3] and [2.1], the data fusion module, the event correlation services module, or both, perform(s) the “*cleaning, classifying, and integrating*” actions. Jakobson explains that the event correlation services module performs these actions on data processed by the signal fusion module (part of the *sensor network abstraction unit*).



Jakobson, FIG. 7 (annotated)

230. Further, Jakobson explains that the event correlation engine uses instantiations of rules to execute the various event correlation functions. One or more of these rules can also be mapped to the claimed “*sensed data analyzer*,” as illustrated below.



sensed data analyzer includes one or more rule modules or engines within the event correlation module.

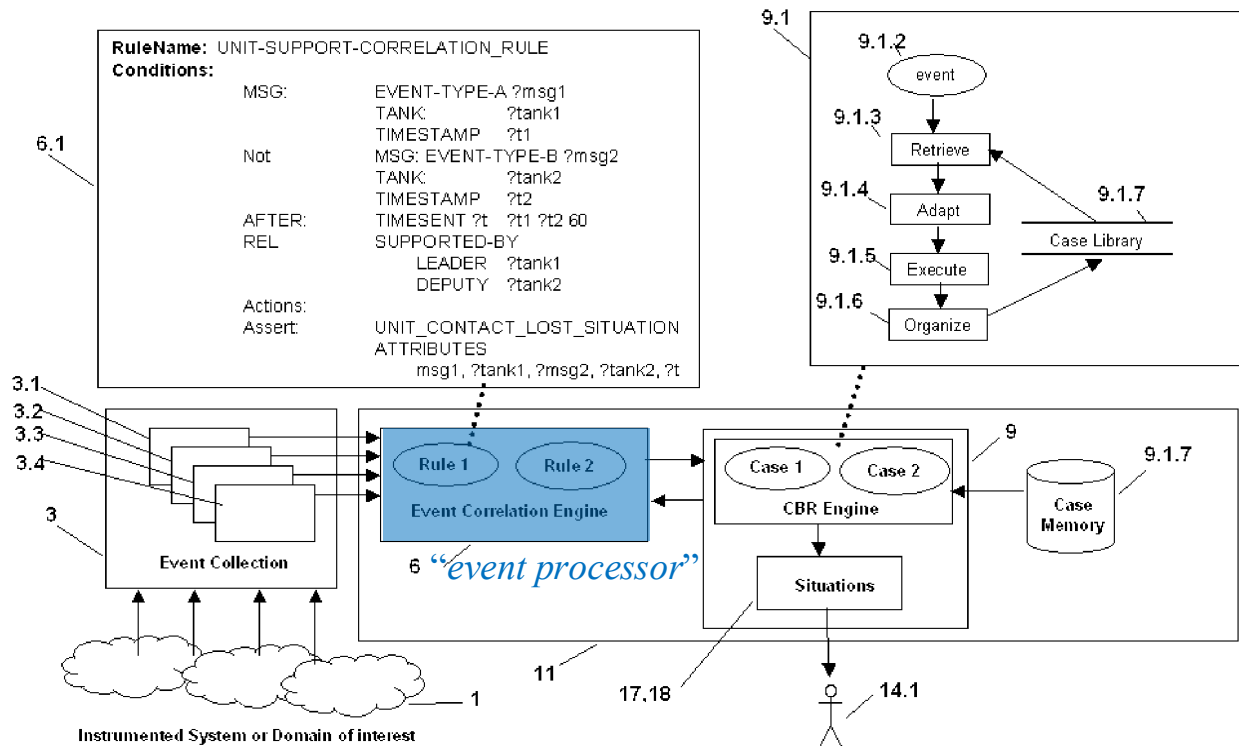
Jakobson, FIG. 6 (partial annotated)

231. Thus, the data fusion services and the event correlation module (correlation services), or one or more rule modules within the event correlation module, discloses *the sensed data analyzer for performing an analysis operation to clean, classify and integrate the sensed data processed by the sensor network abstraction unit.*

[4.1] *an event processor for generating the conditional events including notification and alarm according to a rule set by the application program and managing the generated events; and*

232. As explained above for limitations [1.4] and [2.1], Jakobson's event correlation services/modules create higher-level events based on lower-level events, thresholds, Boolean operators, and conditional predicates, which discloses "*generating the conditional events.*" See Jakobson, [0019] (inferring higher-level

events), [0188] (generating events by thresholding and event clustering based on Boolean operators and conditional terms); [0175] (same), [0176] (thresholding).



Jakobson, FIG. 6 (annotated)

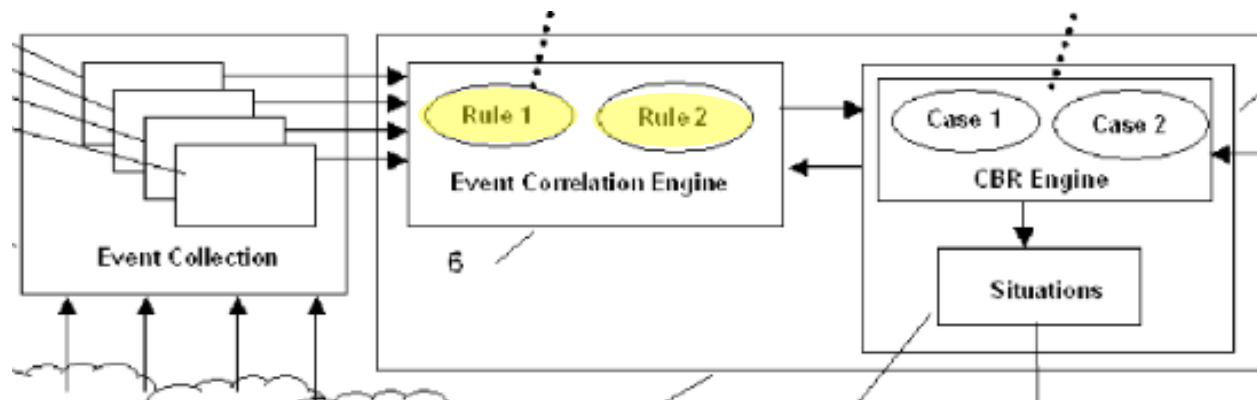
233. As explained above for limitation [2.1], the higher level events created by event correlation include *notification[s]* and *alarm[s]* about a condition (e.g., “power-failure at position 8267,” “radioactivity level = x,” “sighting of vehicle by observer [x],” “UNIT_CONTACT_LOST_SITUATION”). See Jakobson, FIG. 2, [0176], [0191], [0012].

234. Jakobson explains that these higher-level events, including the notifications of the higher-level status of some condition, are governed by *rules* running on a rule engine at the event correlation module:

The event correlation engine 6 may use a general-purpose rules engine such as CLIPS (C Language Integrated Production System) to execute **the correlation rules**.

The types of rules used by the event correlation engine 6 include [compression, filtering, suppression, counting, escalation, generalization, etc.].

Jakobson, [0185]-[0187].



Jakobson, FIG. 6 (partial annotated)

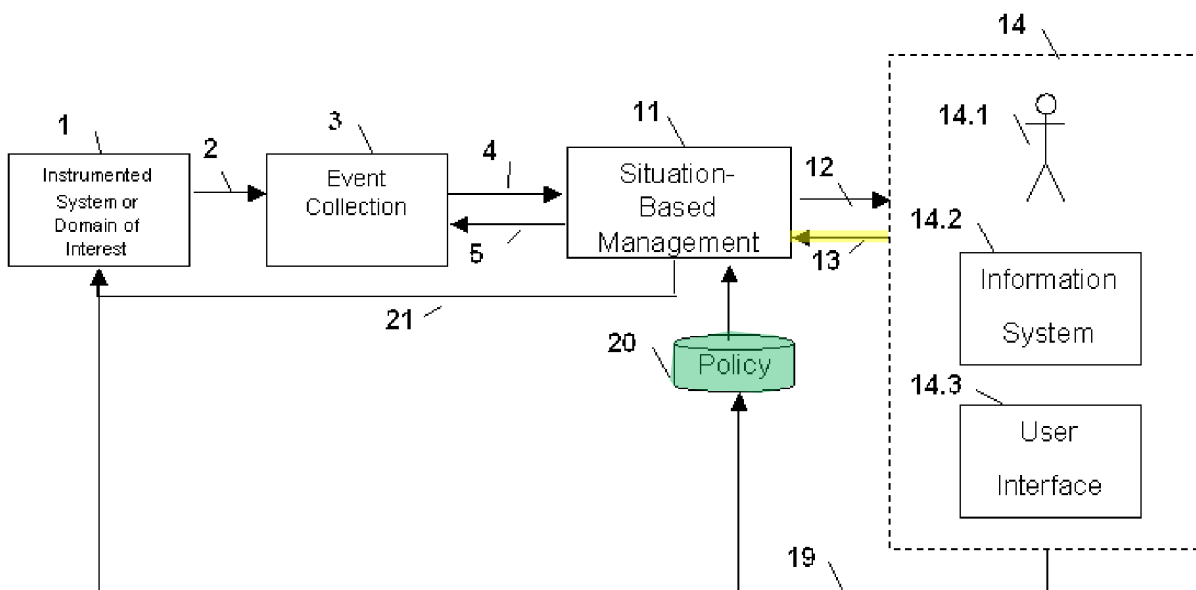
235. Accordingly, the notifications of the properties of the high-level events (*notification and alarm*) are based on *a rule*.

236. Jakobson explains that the rules are executed by the event correlator but does not describe the details of how the event correlation rules are *set*.

However, Jakobson discloses that the situation viewer “may view, manage,

modify, and execute situations” produced by the SBM. Jakobson, [0266].

Jakobson’s situation viewer communicates with the situation-based management through communications channel 13 (yellow), including issuing “remote procedure calls” and “function calls.” Jakobson, [0178].



Jakobson, FIG. 4 (annotated)

237. As also shown in FIG. 4 above, Jakobson explains that the situation viewer 14 “may access, create, and change policies stored in a policy repository 20 that affect the behavior of the situation manager, including the situation evaluation criteria.” Jakobson, [0180]. Thus, Jakobson describes various ways in which the situation viewer and its components are used to control various aspects and policies of SBM, including event correlation and situation management.

238. It would have been obvious for the situation viewer information system or user interface application (*application program*) to configure or “set”

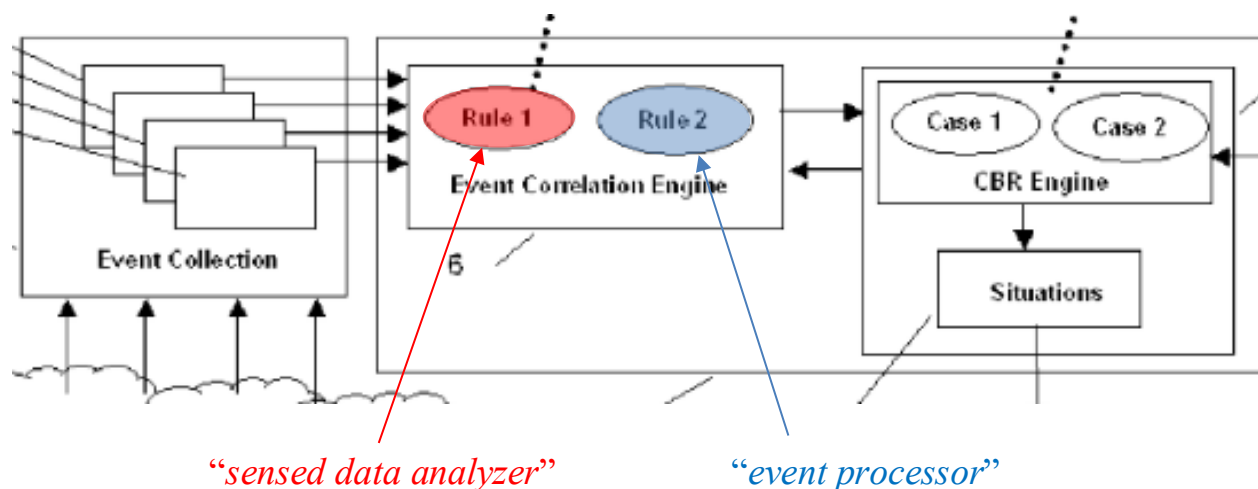
event correlation rules that produce the *conditional events* discussed above. For instance, it would have been obvious for the *application program* to enable the user to control, adjust, or fine tune aspects of event correlation using remote procedure calls or function calls, such as to create or modify (*set*) event correlation rules. For instance, a user viewing the situation reports through the situation viewer user interface may identify issues with the event correlation operations or may wish to customize or optimize event correlation for a given domain or application. Further, a user may understand that certain use cases or circumstances may be more appropriate for more frequent or granular sensor collection than others, or may wish to adjust thresholds, event classification criteria, or other parameters of the event correlation that may result in more sensitive or frequent detection of situations or events that would be passed to the situation viewer.

239. Further, Jakobson explains that the event correlator's objects are "to **direct** the event collection and **event distribution mechanism**; and to permit the situation manager to control or coordinate the EC." Jakobson, [0023]-[0024]. Jakobson explains that the event correlation module is responsible for providing those created events to the situation manager. Jakobson, [0209]. Thus, the event correlator "*manag[es] the generated events.*"

240. As explained above at limitation [4.0], the data fusion module, the event correlation module, or both, may be mapped to the "*sensed data analyzer.*"

In the example or examples in which the event correlation module is mapped to the “*event manager*,” the *sensed data analyzer* can be mapped to one set of rules performing the *cleaning, classifying, and integrating* event correlation actions, and the *event manager* can be mapped to another rule (or set of rules) generating the conditional events. This is illustrated below with respect to FIG. 6.

“*event processor*” and “*sensed data analyzer*” are the rule engines/modules running at the event correlation engine



Jakobson, FIG. 6 (partial annotated)

241. Thus, in this example, the event correlation service, or a rule engine running at the event correlation module, discloses the “*event processor for generating the conditional events including notification and alarm according to a rule set by the application program and managing the generated events.*”

[4.2] a context information manager for sensing and generating the context

aware information, based on the sensed data analyzed by the sensed data analyzer.

242. As explained above for limitation [1.4], in Jakobson's SBM, the situation manager (*context information manager*) generates situations (*context aware information*) and the associated operational context, such as requests for additional/missing information (also *context aware information*). Jakobson, [0297].

243. As similarly explained above for limitation [1.5], the situations and their associated context are generated based on the sensor data (events) provided by the sensor network of the domain and analyzed by the event correlator (*based on the sensed data analyzed by the sensed data analyzer*). Jakobson, [0168]-[0170], [0181].

244. Thus, the situation manager detects and generates (*for sensing and generating...*) situations and their associated operational context (*context aware information*). As similarly explained above for limitation [1.5], the situations and their associated context are generated based on the sensor data (events) provided by the sensor network of the domain and analyzed by the data fusion services and event correlator (*based on the sensed data analyzed by the sensed data analyzer*). Jakobson, [0168]- [0170], [0181].

245. Thus, Jakobson's situation manager (*context information manager*) discloses or renders obvious this limitation.

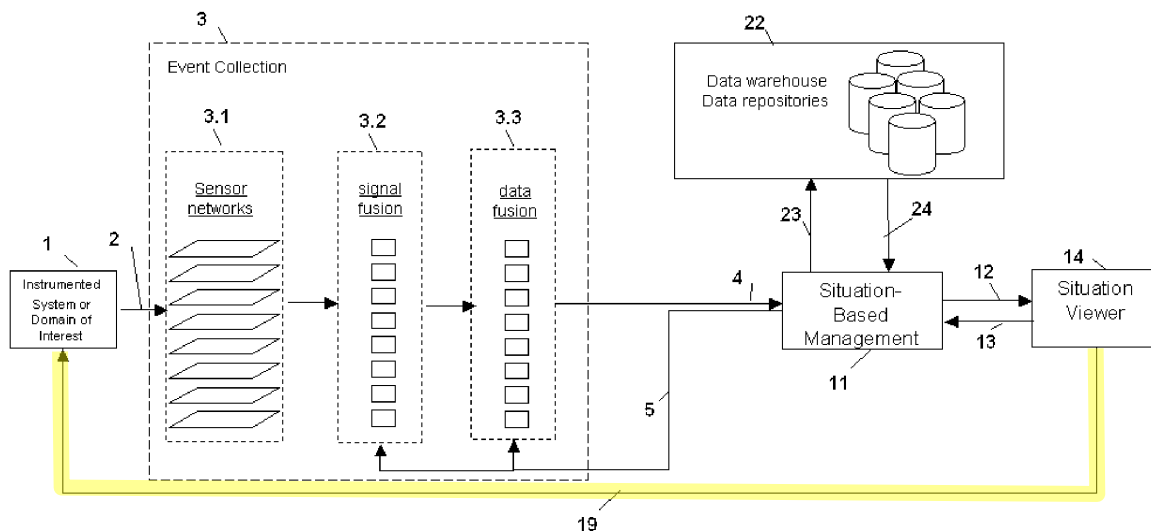
e) Claim 5

[5.0] *The system as recited in claim 4, wherein the service platform management unit comprises: a directory service manager for identifying and searching resources of the sensor network;*

246. Jakobson explains that a “core service” of situation-based management is a “directory” service, which is a “building block” for application services like the situation viewer 14. Jakobson, [0216]. Jakobson further explains that the situation viewer 14 uses a communication channel 19 (highlighted in yellow below) to communicate with the sensors of the domain:

The situation viewer 14 may access, create, and change policies stored in a policy repository 20 that affect the behavior of the situation manager, including the situation evaluation criteria. **The situation viewer 24 may take action on the domain of interest 1 via communications channel 19.**

Jakobson, [0180].



Jakobson, FIG. 5 (annotated)

247. Jakobson also discloses that the situation viewer 14 includes an interface to issue remote procedure calls (e.g., API calls) and request information:

An information system 14.2 may interface to situation-based management 11 through various means for distributed information systems to exchange information such as messages, **remote procedure calls**, and shared databases.

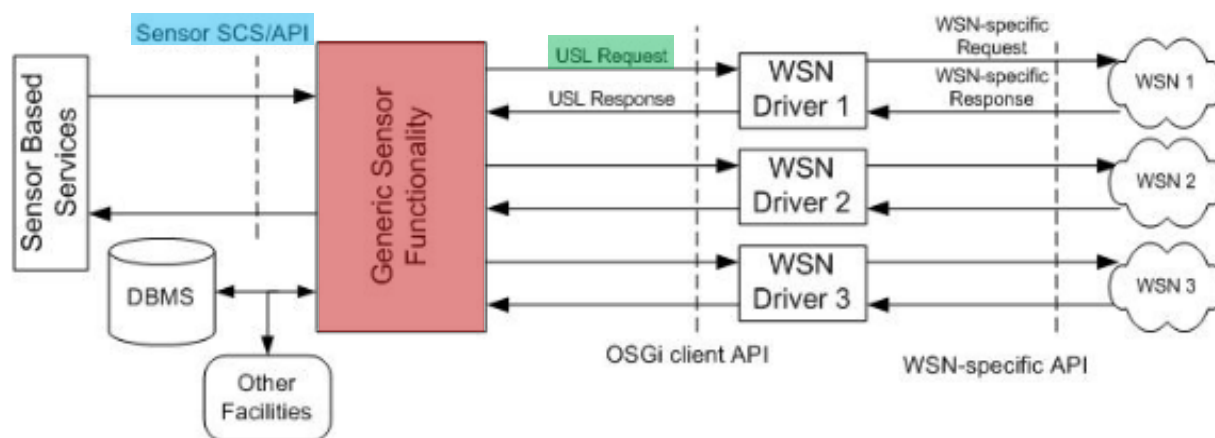
Jakobson, [0178].

248. Thus, Jakobson describes a *directory* service as a building block for SBM middleware, and that the situation viewer 14 can take action on the sensor networks and situation management module 11 by remote procedure calls and other message exchanges.

249. Likewise, Tsetsos also describes functionalities and protocols to issue commands and request information from sensors, such as sensor type, location, and

status (*identifying and searching resources of the sensor network*). See Tsetsos, 5-7 (describing USL and GSF commands and requests). For instance, Tsetsos describes implementing application programming interfaces (APIs) to query a sensor network using pre-defined messages based on a Sensor Service Capability Server (SCS) and “Generic Sensor Functionality” (GSF). Tsetsos, 4-6.

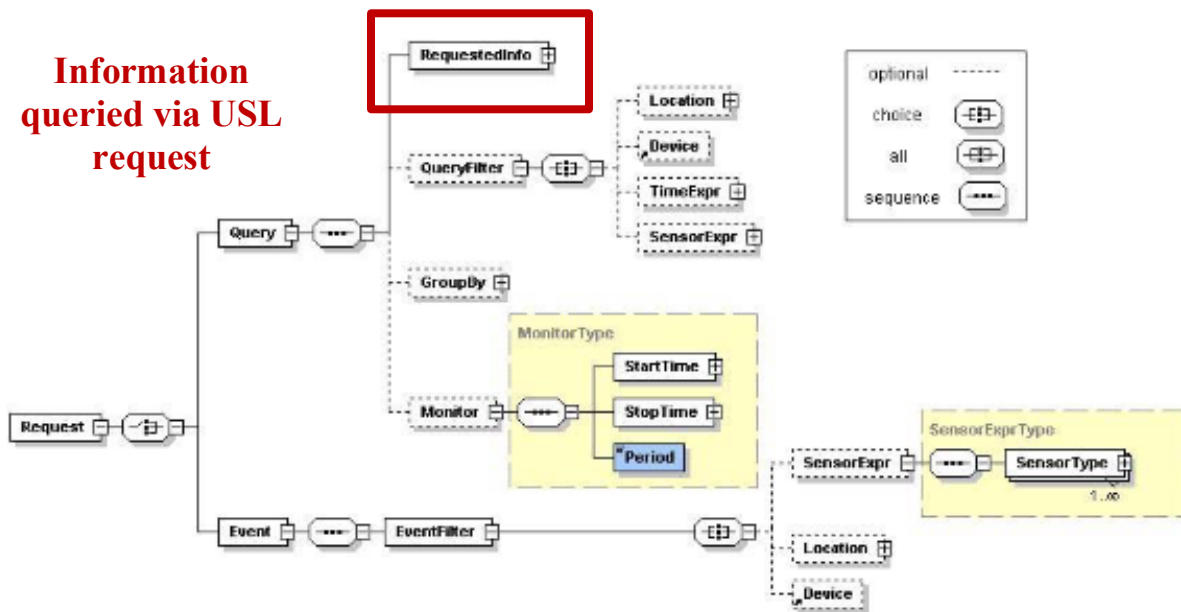
GSF receives SCS commands and issues USL requests



Tsetsos, FIG. 4 (annotated)

250. Tsetsos explains that these APIs can be used to get information about sensors, including *identifying and searching for* specific sensors and associated information. Tsetsos, 6, 7. For instance, the SCS API provide “a common way for the interaction between the service logic and the actual sensor network,” and includes features for quality of service and “discovery.” Tsetsos, 6. Further, Tsetsos explains that the GSF API provides “catalogs for discovery of available

sensor networks.” Tsetsos, 6. Further, Tsetsos explains using “Unified Sensor Language” (USL) requests to obtain information about the sensors in the network (*for identifying and searching resources of the sensor network*), such as sensor type and location. FIG. 5 of Tsetsos below is a diagram illustrating how information can be obtained through USL requests.



Tsetsos, FIG. 5 (annotated)

251. Tsetsos includes an example USL request that shows how location information, sensor type, and device ID can be requested.

```
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance" xsi:noNamespaceSchemaLocation="  
USLRequest.xsd" ID="123456">  
  <Query>  
    <RequestedInfo location="false" time-instance="false"  
      device="true" sensor="true" time-duration="false">  
      <SensorList>  
        <SensorType function="NoFunction"  
          type="temperature"/>  
      </SensorList>  
    </RequestedInfo>  
    <QueryFilter>  
      <Device ID="543"/>  
    </QueryFilter>  
  </Query>  
</Request>
```

Tsetsos, Listing 1, p. 7 (annotated)

252. It would have been obvious for the middleware 25 to include a GSF module (*directory service manager*), like the one disclosed in Tsetsos, to allow a user to query the status of the sensor networks and other sensor information (*for identifying and searching resources of the sensor network*). See *supra*, § V.C.

253. In that regard, Jakobson explains that the situation manager may determine that certain properties of a situation are missing or incomplete, and generate requests for additional information to increase the confidence level of the situation. See Jakobson, [0282], [0298]-[0300].

254. Thus, Jakobson renders this limitation obvious, alone or in view of Tsetsos.

[5.1] a service process manager for scheduling and managing an operation

sequence of the sensed data analyzer, the event processor, the context information manager, and the directory service manager in order to obtain a desired information service according to the application program; and

255. Jakobson explains that its event correlation and situation management techniques include sequences of operations. See Jakobson, [0075] (situation management steps are “sequential and mutually temporally non-overlapping”), [0273]-[0278] (FIG. 15 shows “sequence of steps...in time order” which are “non-overlapping”), [0279]-[0280] (FIGS. 16 and 17 show “sequences of steps” and “sequences of operations”).

256. Jakobson explains that operational sequences for situation management “might be realized on a multiprocessor computer or using distributed processing in a data processing network.” Jakobson, [0280]. Further, parallel sequences of operations “might be implemented in separate computer operating system tasks or processes, in a single process with one thread per activity, or using some other computer system concurrency paradigm.” Jakobson, [0280]. Accordingly, Jakobson discloses that various SBM processes, including situation management, are performed *operation sequence[s]*, and controlled by computer processing circuitry and software (e.g., computer operating system process(es)).

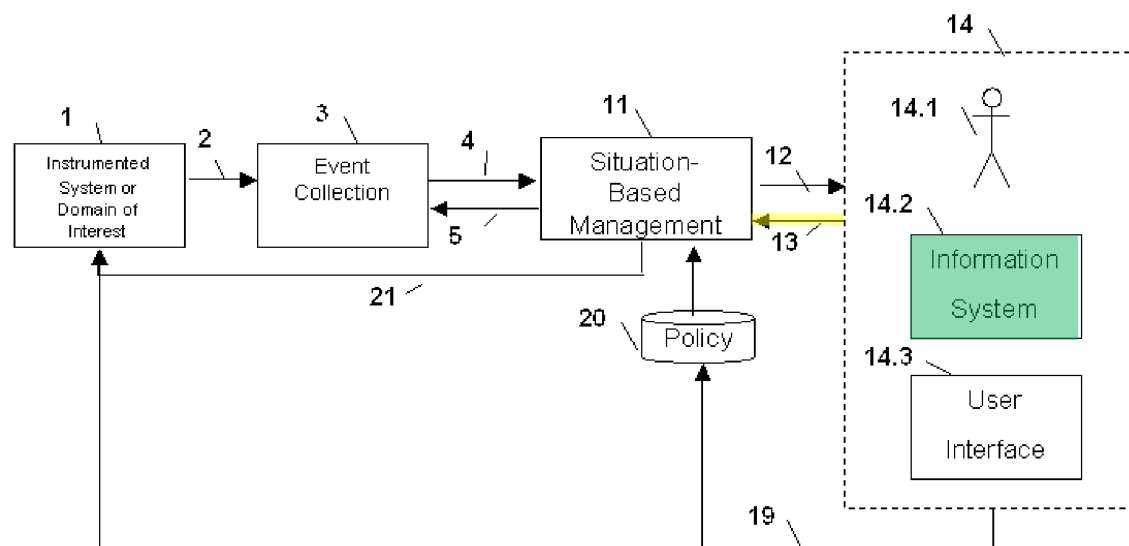
257. Although Jakobson does not discuss at length the structure of the software modules or paradigms for managing and controlling the operation sequences for SBM, it would have been obvious for the SBM middleware to

include a software module for controlling the sequences of steps of *the sensed data analyzer, the event processor, the context information manager, and the directory service manager*.

258. Jakobson's situation viewer 14 includes an information system 14.2 that communicates with the situation-based management 11, including issuing "remote procedure calls," using "polling," or using "function calls":

An information system 14.2 may **interface to situation-based management 11** through various means for distributed information systems to exchange information such as messages, **remote procedure calls**, and shared databases. An information system 14.2 may be provided a situation, situation report, an extract of a situation, an abstract or summary of a situation or a description of a situation from situation-based management 11 by methods such as **polling**, subscription, retrieval, or **function call**.

Jakobson, [0178].



Jakobson, FIG. 4 (annotated)

259. Further, Jakobson explains that the situation viewer 14 communicates with the situation-based management to control various aspects of situation management:

In FIG. 4, the situation viewer 14 may also operate on situation-based management 11 via communications channel 13. These operations include **changing properties of situations, creating new situations, modifying the knowledge base of situation-based management, and modifying the ontology of the situation manager.** The situation viewer 14 may access, create, and change policies stored in a policy repository 20 **that affect the behavior of the situation manager, including the situation evaluation criteria.** The situation viewer 24 may take action on the domain of interest 1 via communications channel 19.

Jakobson, [0180].

260. Thus, Jakobson discloses that the situation viewer can issue commands and set policies for (*for...managing...*) situation-based management operations (*an operation sequence*).

261. It would have been obvious for the SBM middleware 25 to include a module (*service process manager*) similar to the policy repository 20 to manage the sequence of operations used by the various modules of SBM. As explained above, Jakobson explains that the sequences can be inter-dependent or non-overlapping. Jakobson, [0280]. It would have been obvious for policy repository to specify the sequence of operations to ensure that the operations occur at the correct times.

262. For instance, because the situation viewer 14 serves as the interface for the user, and includes functionality to control aspects of the situation-based management, it would have been obvious for the information system 14.2 to enable configurations and user controls to manage, adjust, and fine tune the situation-based management based on the situation reports the user views.

263. Because the data fusion services, the event correlator, the situation manager (*the sensed data analyzer, the event processor, the context information manager*) are providing events and situations as part of *the information service* provided to the situation viewer's information system or user interface program (*application program*), it would have been obvious and straightforward to provide

the user access to operational sequences and logic to adjust the event collection, event correlation, and situation management functions.

264. Jakobson explains that case-based reasoning and situation management can be improved over time through “human-guided learning.” Jakobson, [0053]. Further, Jakobson explains that the communications channel for communicating between the situation manager and EC, for instance, “may be performed or aided by human intervention, where a human operator takes output from the SM and provides the output as input to the EC.” Jakobson, [0074].

265. Providing user-configurable settings and control was a common and effective approach to allow users to adapt a service to the unique circumstances and hardware capabilities (or limitations) involved with the sensor network. A POSITA would have appreciated that this would be particularly useful in a heterogeneous sensor network, and to identify situations that may be very specific to a given setting and application, which Jakobson’s SBM is intended to provide. Further, this would have allowed for SBM procedures to improve over time, based on human input. Jakobson, [0052].

266. It would also have been obvious for this policy repository and management to include policies related to collecting resource information (e.g., sensor metadata) provided by the sensors, for scanning the sensor networks for faults and lost communications, for handling calls for additional information or

sensor statuses, and other operational sequences of *the directory service manager* described above at limitation [5.0].

267. Thus, Jakobson renders this limitation obvious.

[5.2] *an application program interface for providing the application program with the results processed by the sensor network monitor, the sensed data analyzer, the event processor, the context information manager, and the directory service manager in an information service form suitable for the application program.*

268. As explained above at limitation [5.1], Jakobson discloses using a “remote procedure call” to “interface to” SBM to trigger or request situation reports:

An information system 14.2 may **interface to situation-based management 11** through various means for distributed information systems to exchange information such as messages, **remote procedure calls**, and shared databases. An information system 14.2 may be provided a situation, situation report, an extract of a situation, an abstract or summary of a situation or a description of a situation from situation-based management 11 by methods such as polling, subscription, retrieval, or function call.

Jakobson, [0178].

269. Further, as explained above at limitation [5.0], Tsetsos describes using *application program interfaces* (APIs) and Generic Sensor Functionality (GSF) to query information from sensor networks. *See* Tsetsos, 5-7. It would have been obvious to use an API (*application program interface*) to trigger or configure the

provision of situation reports to the information system or user interface program (*for providing the application program...*), as Tsetsos does. APIs like those exploiting the SCS and GSF in Tsetsos provide a uniform and scalable mechanism to control how sensor data and information is obtained and processed.

270. Jakobson discloses modifying the format (*in an information service form suitable for the application program*) of a situation report (*results processed by the sensor network monitor, the sensed data analyzer, the event processor, the context information manager, and the directory service manager*) before delivering the report to the information system 14.2:

The situation, situation report, extract of a situation, abstract or summary of a situation, or description of a situation may be **converted to a different format prior to or during the step of providing it to the information system 14.2.**

Jakobson, [0178].

271. Further, referring to FIG. 7, Jakobson describes using software modules such as Data Adaptation services and Presentation services 35 to control the provision of situation information to the situation viewer:

Other services available for application use include...(ii) Data Adaptation 42 to perform data and knowledge translation functions, (iii) Security 38, (iv) **Presentation 35 to support the user interface.**

Jakobson, [0218].

272. To the extent that the “*results processed by the sensor network monitor, the sensed data analyzer, the event processor, the context information manager, and the directory service manager*” requires separate or individual “results” from each respective component of the *USN middleware*, Jakobson discloses that the situation viewer is provided with not only situation reports, but also “events:”

In FIG. 4, the user interface 14.3 provides at least one modality by which situations, **situation reports**, extracts of situations, descriptions of situations, **and events** are presented.
Jakobson, [0179].

273. Thus, Jakobson discloses presenting results processed by multiple components in the SBM hierarchy, including both situations and events. As explained above with respect to limitation [1.4], “events” can include cleaned, classified, and integrated sensor data, conditional events, situations, and their associated context. It would have been obvious for Jakobson’s situation viewer API to provide the “*results processed*” by the various modules and submodules of the SBM middleware discussed above, including topological information from the topology services (*sensor network monitor*), filtered, compressed, and correlated data from the corresponding event correlation rule engines (*sensed data analyzer*), high-level inferred events from the event correlator or situations from the situation manager (*event processor*), context-sensitive information from the corresponding

situation manager rules (*context information manager*), and sensor network resource information from Tsetsos's GSF module (*directory service manager*).

274. As explained above at limitation [5.1], Jakobson's case-based reasoning and situation management can be improved over time through "human-guided learning." Jakobson, [0053]. Further, Jakobson explains that the communications channel for communicating between the situation manager and EC, for instance, "may be performed or aided by human intervention, where a human operator takes output from the SM and provides the output as input to the EC." Jakobson, [0074].

275. Making the information from the various nodes discussed above (*the sensor network monitor, the sensed data analyzer, the event processor, the context information manager, and the directory service manager*) would have been useful for a situation viewer to identify faults or to view more detailed contextual information of a situation to more precisely assess the threat, the likely cause, and potential remedies. For instance, it was common, and is still common, for applications to allow the user/viewer to retrieve and view more granular information, or predicate information to investigate the details of a reported condition. This information would have been useful for a situation viewer to identify faults or to view more detailed contextual information of a situation to more precisely assess the threat, the likely cause, and potential remedies. *See*

Jakobson, [0005] (network information and topology information used by operations technicians to identify failures).

276. Thus, Jakobson contemplates using remote procedure calls (*an application program interface*) to call for situation reports (*results processed...*) that are provided in a format and presentation that is useable by the situation viewer's information system and/or user interface program (*in an information service form suitable for the application program*).

277. Thus, Jakobson discloses this limitation, or renders it obvious alone or in view of Tsetsos.

f) Claim 6

278. Claim 6 recites substantially the same subject matter as claims 1 and 2, except that it claims a “*Ubiquitous Sensor Network (USN) middleware apparatus*,” instead of a *system* comprising a *USN middleware*, as in claim 1.

[6.0] *A Ubiquitous Sensor Network (USN) middleware apparatus for generating an information service based on heterogeneous sensor networks, comprising:*

279. Jakobson renders obvious this limitation for the same reasons explained above for limitations [1.3] (USN middleware) and [1.0] (an information service based on heterogeneous sensor networks).

280. Although limitation [6.0] recites “*for **generating** an information service*,” instead of “*for **providing** an information service*,” (limitation [1.0]), the

analysis provided above for claim 1 applies to ***generating the information service*** as well as ***providing the information service***.

[6.1] *a sensor network abstraction unit for extracting sensed data from a message received from a sensor network;*

281. Jakobson renders obvious this limitation for the same reasons explained above for limitation [2.0].

[6.2] *a sensor network intelligence unit for extracting object and environment information by performing an analysis operation to clean, classify and integrate the sensed data extracted by the sensor network abstraction unit, and generating an event, including notification and alarm, and context aware information from the extracted object and environment information; and*

282. Jakobson renders obvious this limitation for the same reasons explained above for limitation [2.1].

[6.3] *a service platform management unit for providing the event and the context aware information to the application program in a form suitable for the application program.*

283. Jakobson renders obvious this limitation for the same reasons explained above for limitation [2.2].

g) Claim 7

[7.0] *The ubiquitous sensor network middleware apparatus as recited in claim 6, wherein the sensor network abstraction unit comprises: a sensor network interface for extracting the sensed data by referring to a message transfer protocol;*

284. Jakobson and Tsetsos render obvious this limitation for the same reasons explained above for limitation [3.0].

[7.1] *a sensor network interface protocol storage for managing a rule for the*

message transfer protocol; and

285. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [3.1].

[7.2] *a sensor network monitor for monitoring a state of the sensor network, based on the data received from the sensor network interface.*

286. Jakobson and Tsetsos discloses or renders obvious this limitation for the same reasons explained above for limitation [3.2].

h) Claim 8

[8.0] *The ubiquitous sensor network middleware apparatus as recited in claim 7, wherein the sensor network intelligence unit comprises: a sensed data analyzer for performing an analysis operation to clean, classify and integrate the sensed data processed by the sensor network abstraction unit;*

287. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [4.0].

[8.1] *an event processor for generating the event including the knowledge contents according to a rule set by the application program and managing the generated event; and*

288. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [4.1].

[8.2] *a context information manager for sensing and generating the environment condition information, based on the sensed data analyzed by the sensed data analyzer.*

289. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [4.2].

i) Claim 9

[9.0] *The ubiquitous sensor network middleware apparatus as recited in claim 8, wherein the service platform management unit comprises: a directory service manager for identifying and searching resources of the sensor network;*

290. Jakobson and Tsetsos disclose or render obvious this limitation for the same reasons explained above for limitation [5.0].

[9.1] *a service process manager for scheduling and managing an operation order of the sensed data analyzer, the event processor, the context information manager, and the directory service manager in order to obtain a desired information service according to the application program; and*

291. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [5.1].

[9.2] *an application program interface for providing the application program with the results processed by the sensor network monitor, the sensed data analyzer, the event processor, the context information manager, and the directory service manager in an information service form suitable for the application program.*

292. Jakobson and Tsetsos disclose or render obvious this limitation for the same reasons explained above for limitation [5.2].

j) Claim 10

[10.0] *A method for generating an information service based on heterogeneous sensor networks in a ubiquitous sensor network middleware apparatus, comprising:*

293. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.3] and [1.0]. In that regard, the operation steps and sequences used by the event correlation and situation manager modules

of the SBM middleware are *a method for generating an information service based on heterogeneous sensor networks in a ubiquitous sensor network middleware apparatus.*

[10.1] *extracting sensed data and monitoring a state management of a sensor network, based on the extracted sensed data;*

294. Jakobson and Tsetsos render obvious this limitation for the same reasons explained above for limitations [2.0], [3.0], and [3.2].

[10.2] *performing an analysis operation to clean, classify and integrate the extracted sensed data;*

295. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.3], [2.1] and [4.0] related to *cleaning, classifying, and integrating.*

[10.3] *extracting object and environment information from the analyzed sensed data, and generating context aware information based on the extracted object and environment information;*

296. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [2.1].

[10.4] *generating an event including notification event, alarm, and knowledge contents according to a rule set by an application program;*

297. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [2.1] and [4.1] (*an event including notification event and alarm...according to a rule set by an application program*), and [1.4] (*knowledge contents*).

[10.5] *scheduling an operation sequence for providing an information service to the application program; and*

298. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [5.1] (*scheduling an operation sequence*) and [1.5] (*for providing an information service to the application program*).

[10.6] *providing the application program with the information service obtained according to the scheduled operation sequence, the information service including events and context aware information.*

299. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.5] (*providing the application program with the information service*), [5.1] (*obtained according to the scheduled operation sequence*), and [1.4] (*the information service including events and context aware information*).

k) Claim 11

[11.0] *The method as recited in claim 10, wherein the performing an analysis operation to clean, classify and integrate the extracted sensed data analyzes sensed data from a specific single sensor network, or analyzes sensed data from a plurality of sensor networks, or analyzes sensed data from the sensor network in association with predefined legacy data.*

300. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.3], [2.1] and [4.0].

301. Claim 11 merely recites alternative sources for the *sensed data* being analyzed, including sensed data from (1) “*a specific single sensor network*,” (2) “*a plurality of sensor networks*,” or (3) “*the sensor network in association with*

predefined legacy data.” Because these sources are recited in the alternative, the prior art need only disclose or render obvious one of those options. However, a POSITA would have found it obvious to analyze sensed data from any combination of sensor networks. Jakobson contemplates receiving data from multiple domains or sensor networks. Jakobson, [0054], [0184], FIG. 7. Tsetsos similarly shows that the middleware receives data from multiple wireless sensor networks (WSNs). Tsetsos, FIG. 4. A POSITA would have found it obvious for the SBM processes to receive data from any number of sensor networks, including a “specific single sensor network,” or “a plurality of sensor networks,” as this is consistent with Jakobson’s and Tsetsos’s objectives of a scalable, unified approach to collect and analyze heterogeneous sensor data. *See* Jakobson, [0215]; Tsetsos, 6 (describing unified handling of sensor requests and sensor data).

302. Thus, Jakobson discloses this limitation, or at least renders it obvious in view of Tsetsos.

I) Claim 12

[12.0] *The method as recited in claim 10, wherein the providing the application program with the information service obtained according to the scheduled operation sequence, the information service including events and context aware information provides the information service to the application program only one time, or periodically provides the information service during a request period of the application program, or provides the information service at a specific time point when a requirement from the application program is satisfied.*

303. Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.5] and [5.1].

304. Like claim 11 above, claim 12 recites alternative options related to the frequency with which the information service is provided. Claim 12 recites providing the information to the application program (1) “*only **one time**,*” (2) “*periodically...during a request period of the application program,*” or (3) “*at a **specific time point** when a requirement from the application program is satisfied.*”

305. Because these timing elements are recited in the alternative, the prior art need only disclose or render obvious one of those options. However, to the extent Jakobson and Tsetsos do not explicitly disclose it, a POSITA would have found it obvious to provide situation reports to the situation viewer user interface (provide the application program with the information service) according to any of the recited time frames.

306. For instance, Jakobson discloses updating situations “*periodically.*” Jakobson, Abstract. Further, Jakobson discloses using a procedure call or function call to trigger a situation report (*only one time*):

An information system 14.2 may interface to situation-based management 11 through various means for distributed information systems to exchange information such as messages, **remote procedure calls**, and shared databases. An information system 14.2 may be **provided a situation, situation report**, an extract of a situation, an

abstract or summary of a situation or a description of a situation from situation-based management 11 by methods such as polling, subscription, retrieval, **or function call**. The situation, situation report, extract of a situation, abstract or summary of a situation, or description of a situation may be converted to a different format prior to or during the step of providing it to the information system 14.2.

Jakobson, [0178].

307. A POSITA would have found it obvious for the situation reports to be provided according to any suitable timeframe or temporal configuration, including by request or on-command (*only one time*), periodically, or upon the satisfaction of some condition (*at a specific time point when a requirement from the application program is satisfied*).

308. Accordingly, Jakobson discloses or renders obvious this limitation alone, or in combination with Tsetsos.

IX. CONCLUSION

309. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.



Dated: February 21, 2025

Henry Houh