

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC.,
Petitioner,

v.

KAIFI LLC,
Patent Owner.

IPR2025-00624
U.S. Patent No. 8,040,232

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. §312 AND 37 C.F.R. §42.104**

TABLE OF CONTENTS

PETITIONER’S EXHIBIT LIST	4
LISTING OF CHALLENGED CLAIMS.....	6
I. INTRODUCTION	12
II. GROUNDS FOR STANDING.....	13
III. NOTE.....	13
IV. SUMMARY OF THE ’232 PATENT	13
A. Technological Background	13
B. Overview	13
V. PROSECUTION HISTORY	16
VI. LEVEL OF ORDINARY SKILL IN THE ART.....	17
VII. CLAIM CONSTRUCTION	17
VIII. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF	19
IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	19
A. Challenged Claims	19
B. Statutory Grounds for Challenges	19
C. Ground 1: Claims 1-12 are obvious under §103 over Jakobson in view of Tsetsos.....	21
1. Jakobson.....	21
2. Summary of Tsetsos.....	24
3. Claim 1	26

4.	Claim 2.....	63
5.	Claim 3.....	72
6.	Claim 4.....	82
7.	Claim 5.....	91
8.	Claim 6.....	102
9.	Claim 7.....	103
10.	Claim 8.....	103
11.	Claim 9.....	103
12.	Claim 10.....	104
13.	Claim 11	105
14.	Claim 12.....	106
X.	DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	108
A.	No basis for <i>Fintiv</i> denial.....	108
B.	No basis for §325(d) denial.....	109
XI.	CONCLUSION.....	110
XII.	MANDATORY NOTICES	111
A.	Real Party-in-Interest	111
B.	Related Matters.....	111
C.	Lead and Back-up Counsel and Service Information	111
	CERTIFICATE OF WORD COUNT.....	113
	CERTIFICATE OF SERVICE	114

PETITIONER’S EXHIBIT LIST

Ex.1001	U.S. Patent No. 8,040,232
Ex.1002	Prosecution History of U.S. Patent No. 8,040,232
Ex.1003	Declaration of Henry Houh under 37 C.F.R. §1.68
Ex.1004	<i>Curriculum Vitae</i> of Henry Houh
Ex.1005	U.S. Patent Application Publication No. 2005/0222811 to Jakobson et al. (“Jakobson”)
Ex.1006	Commercial Wireless Sensor Networks: Technical and Business Issues, by Tsetsos et al. (“Tsetsos”)
Ex.1007	Newton’s Telecom Dictionary, 22 nd ed. (2007)
Ex.1008	Computer Networks, 4 th ed. (2004), Andrew S. Tanenbaum.
Ex.1009	Declaration of Gordon MacPherson
Ex.1010	EDTX Local Rules 2023
Ex.1011	Docket Control Order, <i>KAIFI LLC v. Amazon.com, Inc. et al</i> , No. 2:24-cv-00542 (EDTX)
Ex.1012	Statistics on District Court Timing (Sept. 30, 2024)
Ex.1013	Interim Procedure for Discretionary Denials in AIA Parallel District Court Litigation, June 21, 2022
Ex.1014	Judge Gilstrap Docket for March 16, 2026
Ex.1015	Preliminary Infringement Contentions, <i>KAIFI LLC v. Amazon.com, Inc. et al</i> , No. 2:24-cv-00542 (EDTX)
Ex.1016	Tomioka Katsumi, “Ubiquitous Sensor Network System,” NEC Global, C&CEX User Forum & iEXPO2005
Ex.1017	Daniele Puccinelli et al., “Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing,” IEEE Circuits and Systems

	Magazine, (3rd Quarter, 2005)
Ex.1018	Irfan A. Essa et al., “Ubiquitous Sensing for Smart and Aware Environments,” IEEE Personal Communications, (October 2000)
Ex.1019	Peter Schramm et al., “A Service Gateway for Networked Sensor Systems,” IEEE Pervasive Computer (1 st Quarter, 2004)
Ex.1020	Gaurav Gupta et al., “Fault-Tolerant Clustering of Wireless Sensor Networks,” IEEE Wireless Communications and Networking (May 2003)

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1.0]	A system for providing an information service based on heterogeneous sensor networks, comprising:
[1.1]	a sensor node constructing a sensor network to sense environment information;
[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
[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,
[1.4]	generating conditional events, context aware information, and knowledge contents from the extracted object and environment information, and
[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.
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;
[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

[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.
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;
[3.1]	a sensor network interface protocol storage for managing a rule for the message transfer protocol; and
[3.2]	a sensor network monitor for monitoring a state of the sensor network, based on the data received from the sensor network interface.
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;
[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
[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.
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;
[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
[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.
Claim 6	
[6.0]	A Ubiquitous Sensor Network (USN) middleware apparatus for generating an information service based on heterogeneous sensor networks, comprising:
[6.1]	a sensor network abstraction unit for extracting sensed data from a message received from a sensor network;
[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
[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.
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;
[7.1]	a sensor network interface protocol storage for managing a rule for the message transfer protocol; and
[7.2]	a sensor network monitor for monitoring a state of the sensor network, based on the data received from the sensor network

	interface.
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;
[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
[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
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;
[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
[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.
Claim 10	
[10.0]	A method for generating an information service based on heterogeneous sensor networks in a ubiquitous sensor network

	middleware apparatus, comprising:
[10.1]	extracting sensed data and monitoring a state management of a sensor network, based on the extracted sensed data;
[10.2]	performing an analysis operation to clean, classify and integrate the extracted sensed data;
[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;
[10.4]	generating an event including notification event, alarm, and knowledge contents according to a rule set by an application program;
[10.5]	scheduling an operation sequence for providing an information service to the application program; and
[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.
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.
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.
--	---

I. INTRODUCTION

U.S. Patent No. 8,040,232 (the “’232 patent,” Ex.1001) relates to a middleware-based system for analyzing “low-level” sensed data from a sensor network to generate “high-level information” for an application program. Ex.1001, 2:39-40, 2:43-48. The “high-level information” includes “conditional events” and “context-aware information” that describe a current state or condition of an environment, such as temperature exceeding a threshold, or a potential equipment failure. Ex.1001, 3:16-17, 4:23-25, 5:54-57, 6:14-33 (Table 1, “low temperature,” “problem in the heating machine”).

These concepts were well-known in the art. For instance, U.S. Patent Application Publication No. 2005/0222811 (“Jakobson,” Ex.1005) describes a middleware-based system for “situation management” that analyzes “low-level” data from a sensor network and generates “higher level” events and situations. Ex.1005, [0010], [0019], [0027]-[0029], [0188]. The “higher level” events and situations describe a current state or condition of a system, such as a sensor reading exceeding a threshold, or a potential disruptive or dangerous situation. Ex.1005, [0010], [0176] (“it is suspect that sensor 123 exceeded threshold”), [0177] (“power disruption,” “chemical release”).

Accordingly, pursuant to 35 U.S.C. §§311, 314(a), and 37 C.F.R. §42.100, Amazon.com, Inc. (“Petitioner”) respectfully requests that the Board review and

cancel as unpatentable under (pre-AIA) 35 U.S.C. §103(a) claims 1-12 (hereinafter, the “Challenged Claims”) of the ’232 patent.

II. GROUNDS FOR STANDING

Petitioner certifies that the ’232 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims. 37 C.F.R. §42.104(a).

III. NOTE

Petitioner cites to exhibits’ original page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

IV. SUMMARY OF THE ’232 PATENT

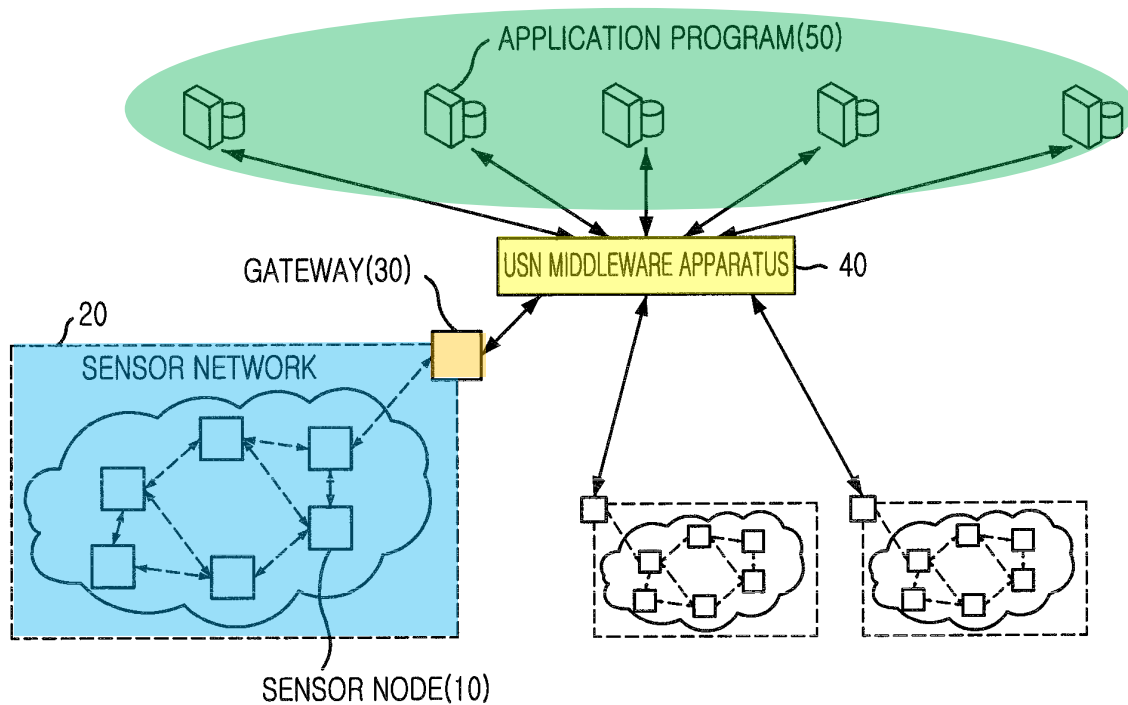
A. Technological Background

The ’232 patent relates to sensor data analysis concepts that were well-known as of the earliest claimed priority date. Dr. Houh’s declaration describes the relevant background technology predating the ’232 patent, which is cited as appropriate below. *See* Ex.1003, ¶¶30-35; Exs.1007, 1008, 1016, 1017, 1018, 1019, 1020.

B. Overview

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.” Ex.1001, 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). Ex.1003, ¶36.

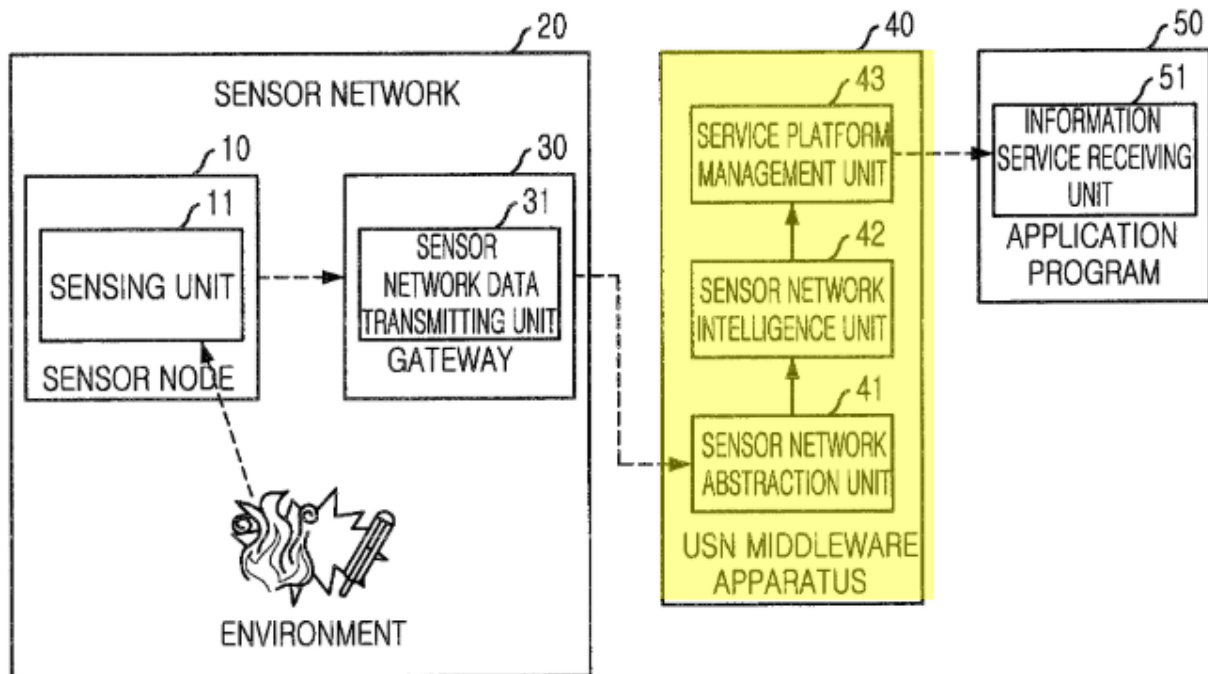


Ex.1001, FIG. 1 (annotated); Ex.1003, ¶36.

The sensor nodes in the sensor network provide sensor data representing environmental conditions like “temperature, humidity, heat, and infrared rays.” Ex.1001, 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).”

Ex.1001, 4:35-37. Ex.1003, ¶37.

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[ing] low-level information, i.e., object properties and environment state information,” and generating “high-level information such as context aware information and conditional events about a wide area.” Ex.1001, 4:20-25, 3:16-17; Ex.1003, ¶38.



Ex.1001, FIG. 2 (annotated); Ex.1003, ¶38.

The '232 patent explains that “low-level information” includes, for instance, “sensed data” collected by sensor networks, and the higher-level information may

include an alarm that temperature readings are below 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”). Ex.1001, 6:1-33 (Table 1); Ex.1003, ¶39.

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. Ex.1003, ¶40.

V. PROSECUTION HISTORY

The ’232 patent issued from U.S. Application filed on December 26, 2006, and claims priority on its face to Korean Patent Application No. 10-2006-0084353, filed on September 1, 2006 (the earliest possible priority date). It is unnecessary to determine whether the ’232 patent is entitled to its earliest possible priority date because the prior art relied upon herein pre-dates the earliest possible priority date.

During a brief prosecution, the Examiner issued a notice of allowance without rejecting the claims. Ex.1002, 270-276. The notice of allowance followed an *Ex Parte Quayle* action and minor amendments to correct “formal matters.” Ex.1002, 246-249. In the statement of reasons for allowance, the Examiner reproduced the entirety of claim 1 as the “allowable subject matter.” Ex.1002, 275-276.

VI. LEVEL OF ORDINARY SKILL IN THE ART

A Person of Ordinary Skill in The Art (“POSITA”) in September 2006 (the earliest possible priority date) would have been someone knowledgeable and familiar with the 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 field of digital networks and data processing software. Additional education could compensate for lack of work experience, and significant experience in the field could substitute for formal education. Ex.1003, ¶¶24-25.

VII. CLAIM CONSTRUCTION

No claim terms require express construction to resolve the issues presented in this Petition.¹ *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). For the purposes of this Petition, Petitioner applies Patent Owner’s apparent position in its Infringement Contentions that no terms invoke §112, ¶6. *See* Ex.1015, 13-264 (no terms identified as invoking §112,

¹ Petitioner does not concede that any term in the challenged claims satisfies §112 or recite patentable subject matter under §101.

¶6²); *see also* 37 C.F.R. §42.104(b)(3) (requiring the petition to set forth only “how the challenged claim is to be construed”); *Hospira, Inc. et al. v. Amgen Inc.*, IPR2021-00528, Paper 7 at 7 (Aug. 17, 2021) (“We determine that Petitioner’s willingness to adopt Patent Owner’s proposed constructions for purposes of this proceeding is sufficient to satisfy the requirements of §42.104(b)(3).”); *ZTE Corp. et al. v. WSOU Invests., LLC*, IPR2021-00698, Paper 10 at 8-10 (Oct. 18, 2021) (declining to apply §112 ¶6 where a claim term did not include the words “means” and Petitioner adopted Patent Owner’s plain and ordinary meaning construction for the purposes of the proceeding).

In the event the Board finds that any term in the Challenged Claims invokes §112, ¶6, for the purposes of this proceeding Petitioner adopts as the corresponding structure a general purpose “computer” running “programs [] stored in a computer-readable recording medium that can execute the programs.” *See* Ex1001, 8:13-17; *see also General Electric Co. v. Vestas Wind Systems A/S*, IPR2018-00928, Paper 9 at 15–16 (Nov. 5, 2018) (“In other words, the rule [37 C.F.R. §42.104(b)(3)] does not prohibit a petitioner from submitting more than one construction, including one that it believes is incorrect, such as a patent owner’s expected construction.”).

² *See* Ex.1010 (EDTX Local Rules), 55.

Further, claims 1 and 6 recite a “Ubiquitous Sensor Network (USN) middleware.” For the purposes of this Petition, Petitioner applies Patent Owner’s apparent understanding that this term should ***not*** be interpreted as proper noun such that only a disclosure of a “Ubiquitous Sensor Network (USN) middleware” would satisfy the claim. *See* Ex.1015, x-x (mapping “AWS IoT Core” to this claim term).

VIII. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims in view of the analysis below.

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Challenged Claims

Petitioner challenges claims 1-12, which correspond to the claims asserted in the plaintiff’s infringement contentions in the co-pending litigation.

B. Statutory Grounds for Challenges

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

U.S. Patent Application Publication No. 2005/0222811 to Jakobson et al. (“**Jakobson**”) titled “Method and Apparatus for Context-Sensitive Event Correlation with External Control in Situation-Based Management,” is a printed

publication that was published on October 6, 2005. Jakobson was filed on April 2, 2005. Jakobson is therefore prior art to the '232 patent under at least pre-AIA 35 U.S.C. §§102(e) and 102(a)-(b).

Commercial Wireless Sensor Networks: Technical and Business Issues, by Tsetsos et al. (“**Tsetsos**”) is a printed publication that was first published in January 2005. As indicated by the declaration of the custodian of records for IEEE (Ex.1009), the article was made publicly available as of the last day of the Second Annual Conference on Wireless On-demand Network Systems and Services, which occurred between January 19-21, 2005, and it was published and made available for public download in the IEEE digital library, IEEE Xplore on January 31, 2005. Ex.1009, ¶¶10, 11; *see also Ericsson, Inc. v. Intellectual Ventures I LLC*, IPR2014-00527, Paper 41 at 11 (May 18, 2015) (“IEEE is a well-known, reputable compiler and publisher of scientific and technical publications, and we take Official Notice that members in the scientific and technical communities who both publish and engage in research rely on the information published on the copyright line of IEEE publications.”); *Valve Corp. v. Ironburg Inventions Ltd.*, 8 F.4th 1364, 1374 (Fed. Cir. 2021) (“where a publication’s purpose is ‘dialogue with the intended audience,’ that purpose indicates public accessibility”). Tsetsos is thus prior art under at least §§102(a)-(b).

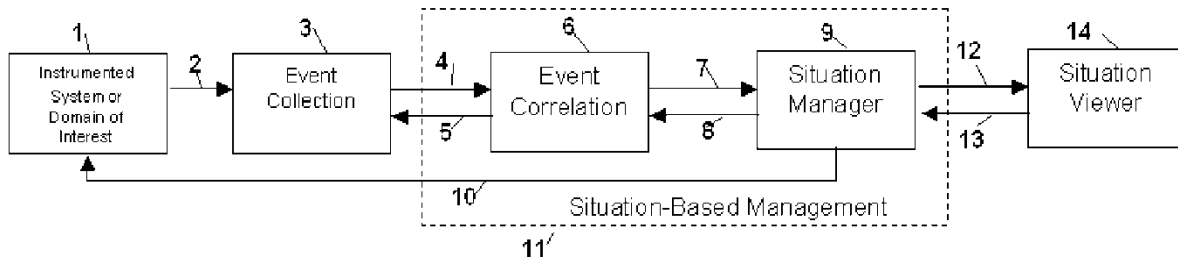
Petitioner’s analysis also cites additional prior art to demonstrate the

background knowledge of a POSITA and to provide contemporaneous context to support Petitioner’s assertions regarding what a POSITA would have understood from the prior art. *See Yeda Research & Dev. Co. v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018).

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

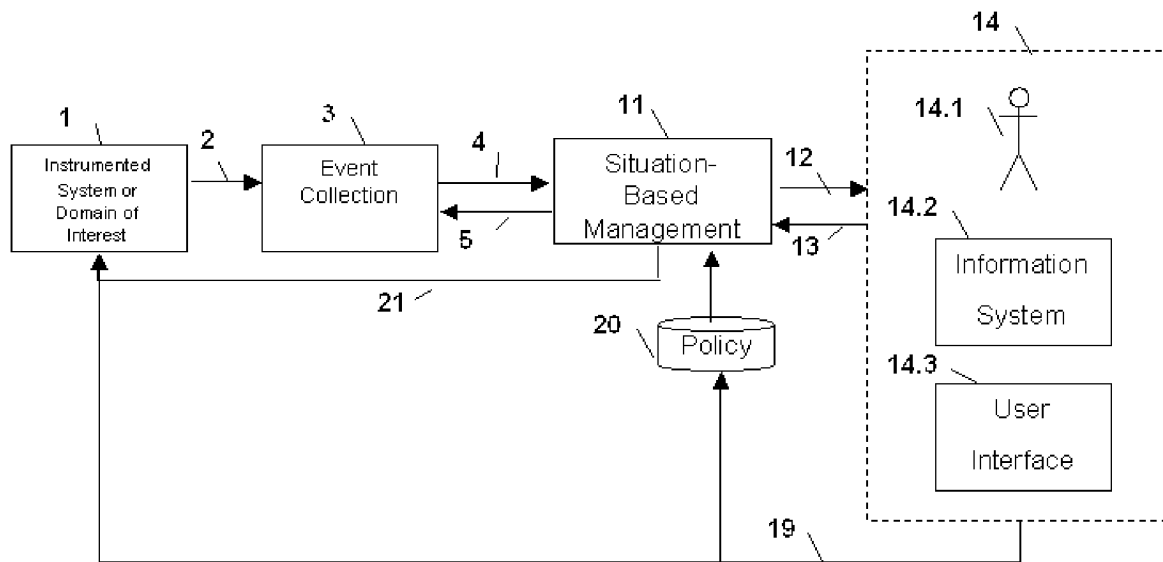
1. Jakobson

Jakobson describes methods and systems for “context-sensitive event correlation” and “situation-based management” (SBM) using data collected from “sensor networks” having “a wide variety of types of sensors.” Ex.1005, Abstract, [0010], [0015], [0181]. Jakobson implements SBM systems using “middleware,” which facilitates “the building of open, scalable, and customizable systems.” Ex.1005, [0215], [0216]. SBM techniques include collecting and processing sensor data from a domain to identify higher-level “events” and “situations.” Ex.1005, Abstract, [0019], [0188]. FIG. 1 of Jakobson shows an example situation management process flow including functional modules for collecting sensor data and using that data to identify events and situations. Ex.1003, ¶46



Ex.1005, FIG. 1

The “inferred” or “correlated” events and situations convey higher level, useful information about the domain as compared to basic sensor readings. *See* Ex.1005, [0019] (e.g., a “root-cause”). For instance, a collection of low-level sensor data representing sound, pressure, temperature are combined, correlated, and processed through a multi-level process to identify “high level” events and situations like “power disruption,” or “chemical release.” Ex.1005, [0010]-[0013], [0177]. Situation reports are provided to users via a “user interface 14.3.” Ex.1005, [0178]-[0179]; Ex.1003, ¶47.



Ex.1005, FIG. 4.

Jakobson refers to a broad range of sensor-based information as “events.” Ex.1005, [0012]. Events can include raw sensor data, filtered and consolidated sensor readings, and higher-level descriptions of the state of the domain. Ex.1005, [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 filtering, compressing, counting, and other conditional processes to infer higher-level “correlated” events. Ex.1003, ¶¶48-52.

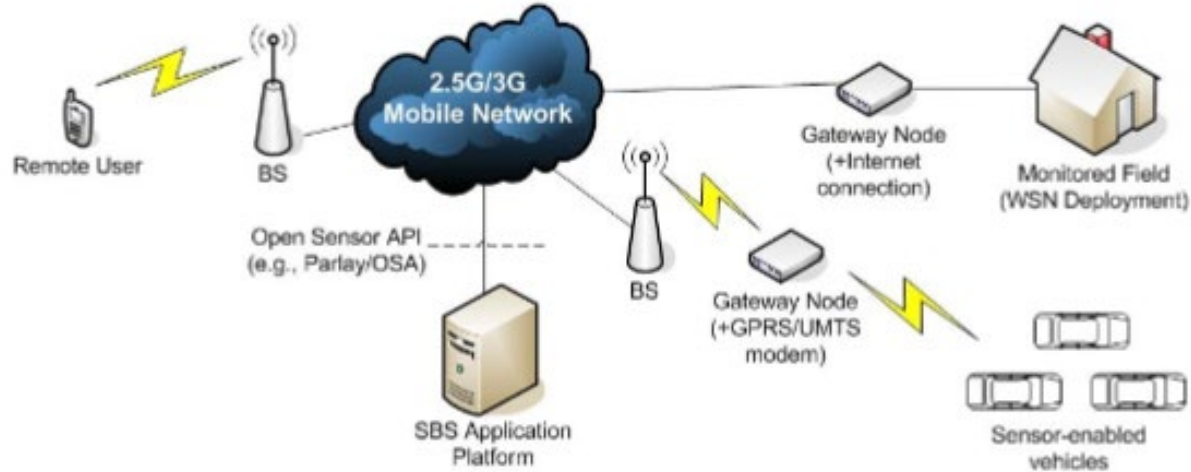
Jakobson is somewhat limited in its discussion of the interfaces between the sensor networks, the middleware, and the user interface because such interfaces were well-known. Ex.1003, ¶¶55-60. Other references, such as Tsetsos, provide

those details. Ex.1003 ¶¶59-60, 30-35 (Technical background explaining known and commonly-used technologies, including gateways, communication protocols, and APIs).

2. Summary of Tsetsos

Like the '232 patent and Jakobson, Tsetsos describes using middleware to communicate with heterogenous sensor networks. Ex.1006, 5. Tsetsos supplies additional implementation details regarding the interfaces and communications between sensor networks, middleware, and third-party applications that were well-known in the art. Tsetsos 3, 5, 6; Ex.1003 ¶¶53-54.

Tsetsos relates to “sensor network” applications that facilitate smart, “context-aware” services. Ex.1006, Abstract, 1. Tsetsos describes using gateway devices, middleware, and application program interfaces (APIs) to manage heterogeneous sensor networks in a unified form. Ex.1006, 3 (gateway nodes are “the connecting link between the sensor network and the WAN”), 5 (discussing the need for “middleware” given the “heterogeneity” of wireless sensor networks). FIG. 1 illustrates gateway nodes collecting and transmitting data from wireless sensor networks (WSNs) to the Internet. Ex.1003 ¶53.



Ex.1006, FIG. 1.

Tsetsos describes using middleware to provide “unified handling” of heterogeneous WSNs. Ex.1006, 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. Ex.1006, 5. These protocols specify APIs that provide developers with a broad set of “network functionality (user interaction, mobility management, user terminal capabilities, location, etc.).” Ex.1006, 5; Ex.1003 ¶54.

In sum, a POSITA, when considering the teachings of Jakobson, would have also considered the teachings of Tsetsos. Ex.1003, ¶¶55-60. 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. Ex.1005, [0014]-[0015], [0215]-[0216]; Ex.1006, 5, 6, 8; Ex.1001, Abstract, 1:22-

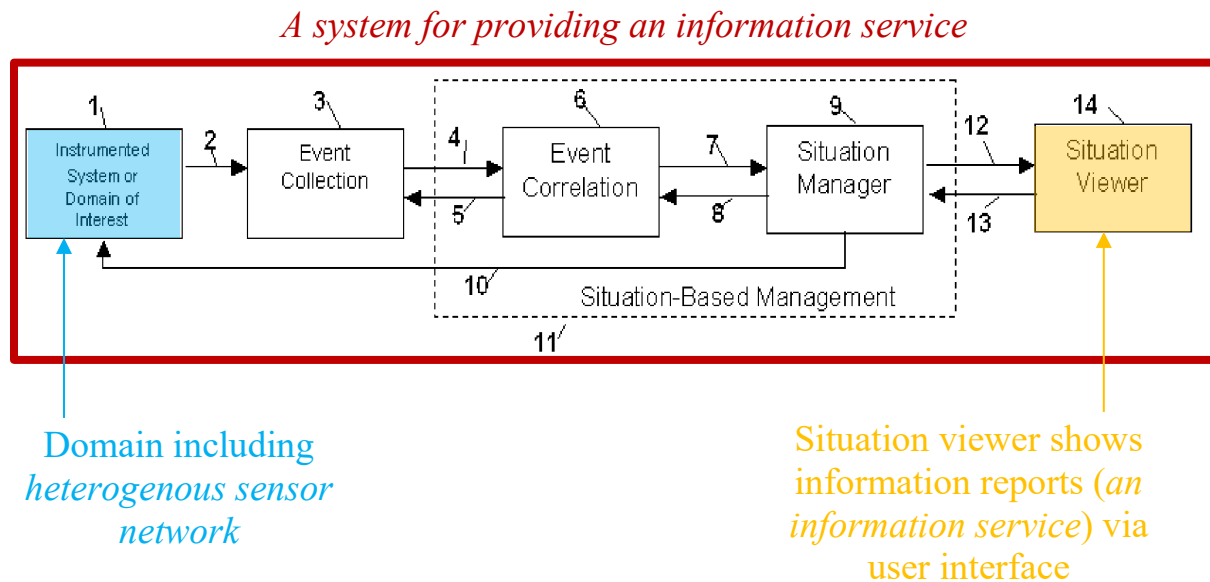
32; Ex.1003, ¶55. Specific motivations for utilizing various elements of Tsetsos in the context of Jakobson’s SBM system are addressed in the below claim mapping.

3. Claim 1

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

Jakobson discloses a *system* for situation-based management (SBM).

Ex.1005, Abstract, [0009]. Jakobson explains that its SBM receives data from “sensor networks” and outputs information reports (*an information service*) to users via a “situation viewer” that includes a “user interface.” Ex.1005, [0181], [0178], FIG. 4. Ex.1003 ¶¶61-62.

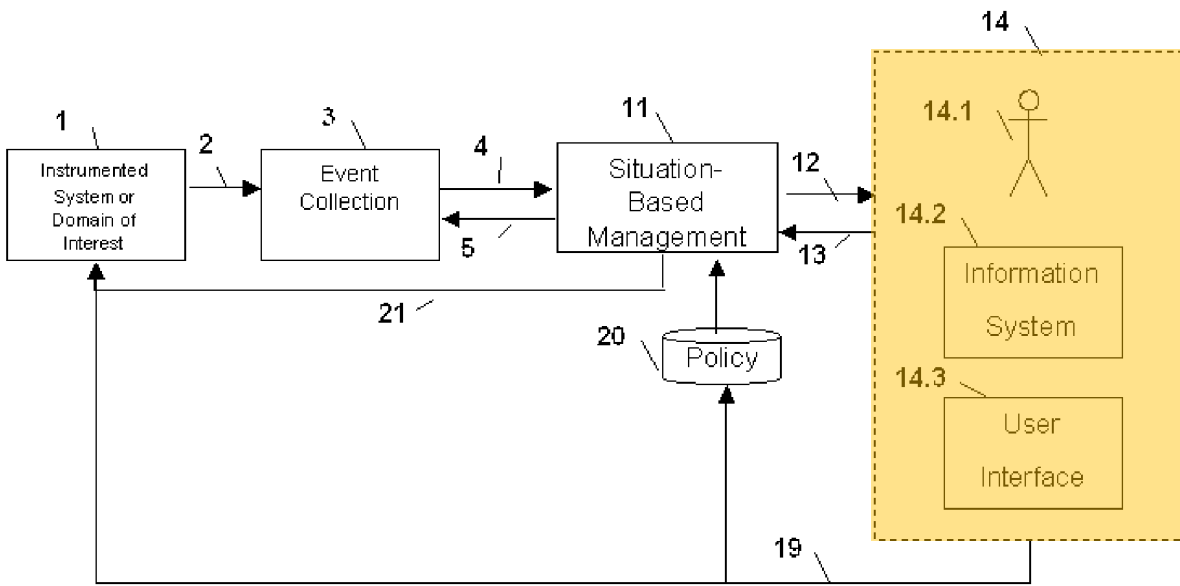


Ex.1005, FIG. 1 (annotated); Ex.1003, ¶62.

With respect to the “sensor networks,” Jakobson explains that SBM system receives sensor information from a “domain” having “**sensors and agents** which

communicate through **sensor networks**.” Ex.1005, [0181]. The sensors may provide data “in digital or analog format.” Ex.1005, [0181]. Jakobson explains that the sensor data are provided by “**a wide variety of types of sensors**.” Ex.1005, [0014]-[0015]. These various (*heterogenous*) types of sensors represent a variety of environmental conditions and characteristics, including “audio, pressure, or temperature.” Ex.1005, [0010]. Thus, Jakobson’s SBM system (*system for providing an information service*) is based on *heterogeneous sensor networks*. Ex.1003 ¶¶61, 72.

With respect to the information reports, Jakobson’s SBM includes outputting reports, notifications, and other information (*information service*) to a user via a “situation viewer 14” (orange, FIG. 4 below) that can include “an information system 14.2 or a user interface 14.3.” Ex.1005, [0178]. Ex.1003 ¶¶69-71.



Ex.1005, FIG. 4 (annotated); Ex.1003, ¶69.

Thus, Jakobson's SBM system for providing situation reports based on data from heterogeneous sensor networks renders obvious *a system for providing an information service based on heterogeneous sensor networks*.

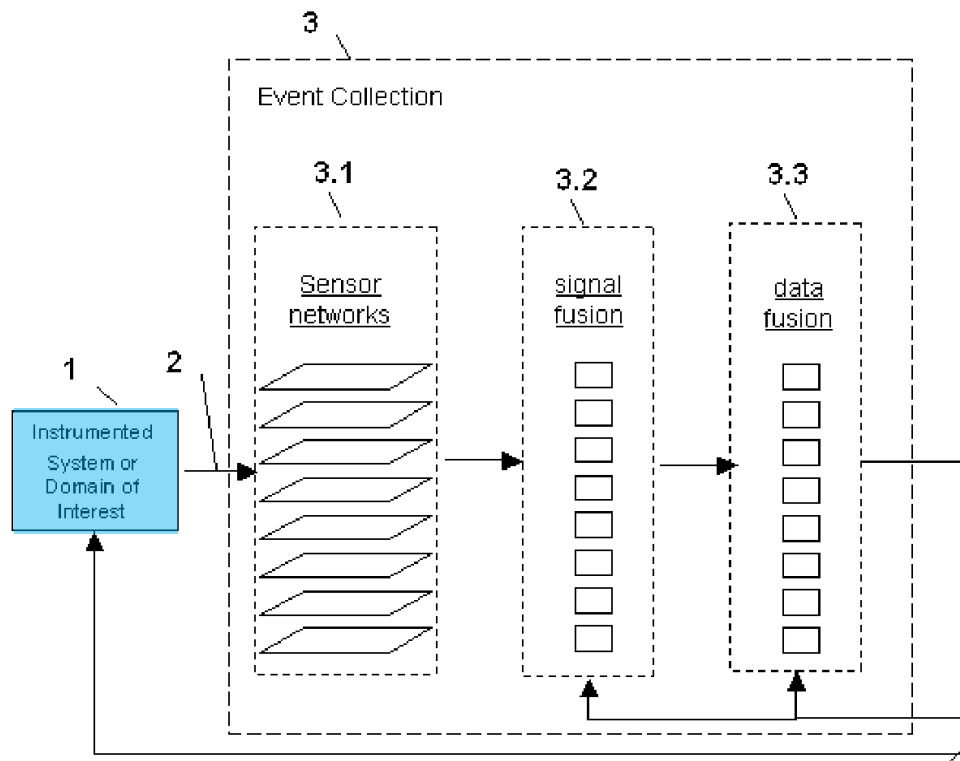
* * *

The analysis below variously refers to features shown in Jakobson's FIGS. 1-8, each of which highlights different aspects of Jakobson's SBM system with varying levels of detail. The essential characteristics of Jakobson's SBM system do not depend upon a specific software architecture or framework, and thus FIGS. 1-8 provide various example implementation options (e.g., flow diagrams, system diagrams, software frameworks). Ex.1005, [0325]-[0326], [0138]-[0145], FIGS. 1-8 (reusing reference numerals across figures); Ex.1003, ¶¶63-68. The SBM

functionalities and modules of Jakobson's system do not require or depend on any specific software architecture or framework. The various implementation options in FIGS. 1-8 leverage standardized and well-known software architectures, such as "standard middleware services or components." Ex.1005, [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.); Ex.1003, ¶67. Thus, the below claim mapping is not predicated upon a combination of embodiments. A POSITA, however, would nevertheless find it obvious to combine aspects of Jakobson's implementations as contemplated by Jakobson. Ex.1005, [0325] ("As will be apparent to those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof."); Ex.1003, ¶¶57-68.

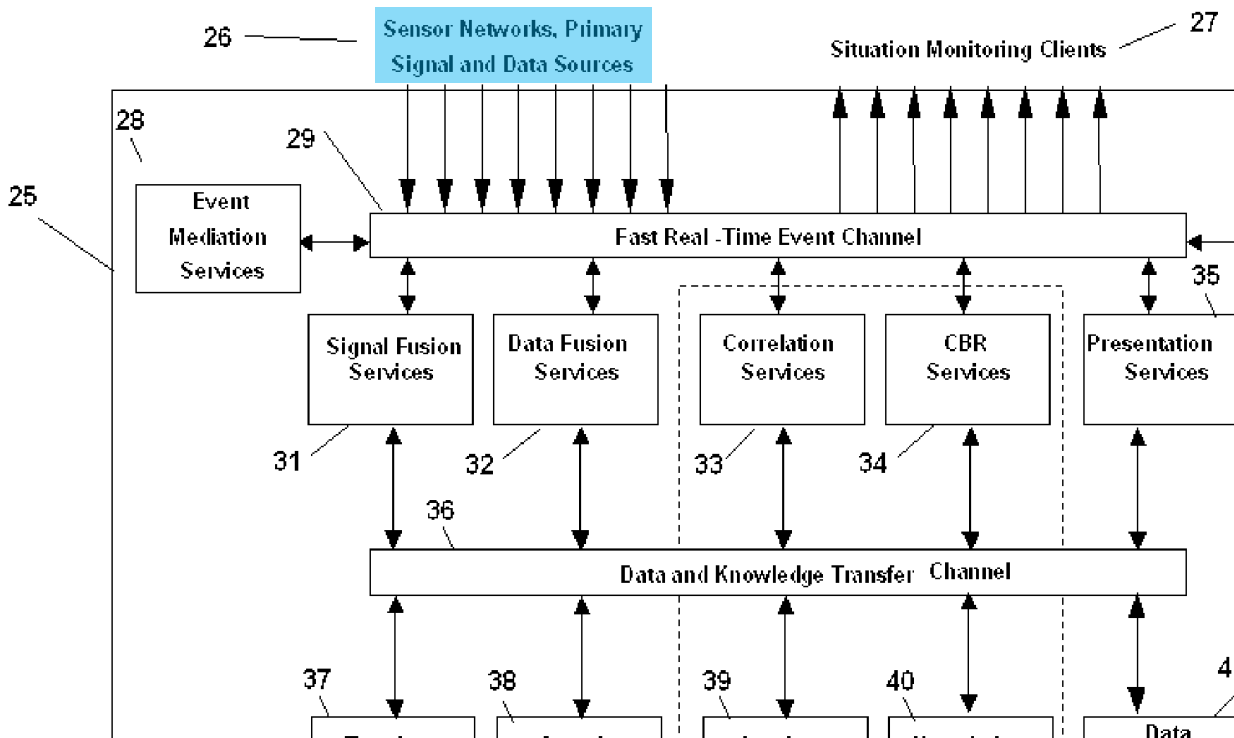
[1.1] *a sensor node constructing a sensor network to sense environment information*

First, Jakobson's "domain 1 is instrumented with sensors and agents which communicate through sensor networks to provide sensor information." Ex.1005, [0181]. Each sensor instrumented at the domain is *a sensor node*. FIG. 5 of Jakobson shows the domain instrumented with sensors (blue).



Ex.1005, FIG. 5 (partial annotated); Ex.1003, ¶74.

Consistent with the description of FIG. 5, Jakobson's FIG. 7 illustrates "sensor networks" (blue) which provide sensor data to the SBM middleware 25. Ex.1005, [0216].



Ex.1005, FIG. 7 (partial annotated); Ex.1003, ¶75.

The *sensor nodes* instrumented at the domain form 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), and/or a sensor gateway (e.g., gateway of Tsetsos, discussed below for limitation [1.2]). Ex.1003, ¶¶74-76.

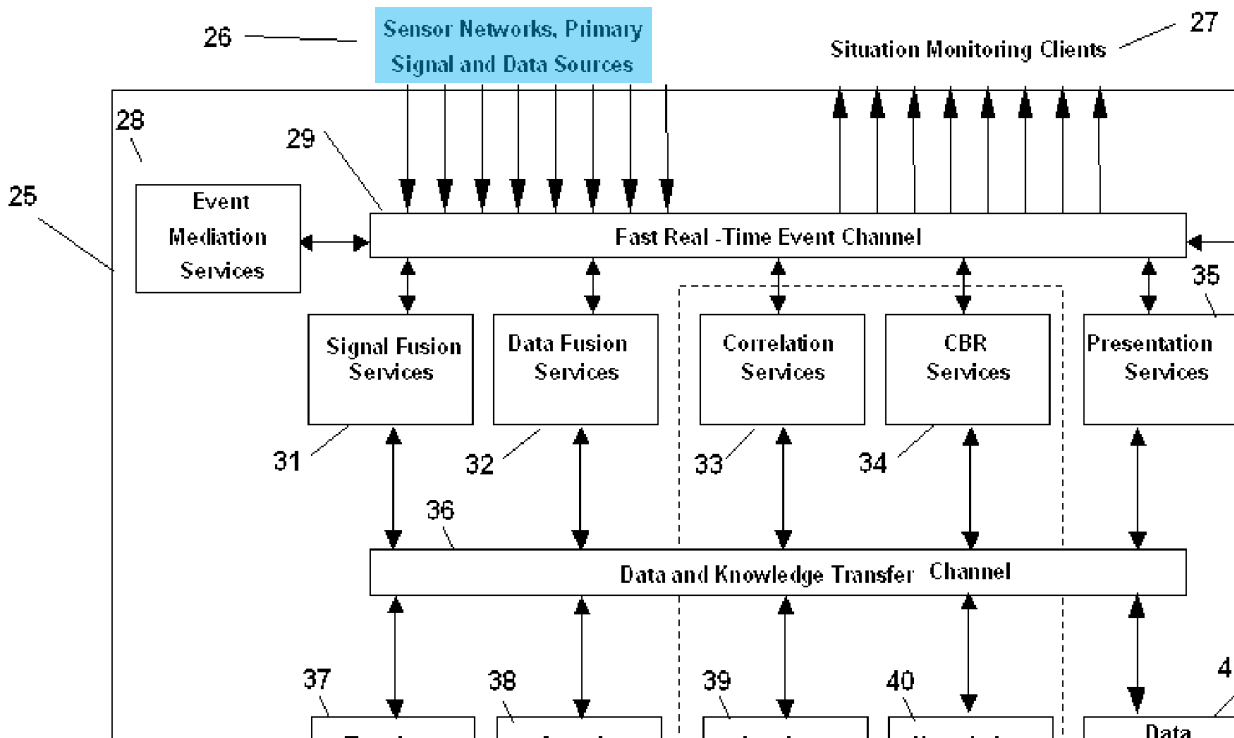
Second, Jakobson explains that sensors collect *environment information* from “real-time sources including sensors on field equipment.” Ex.1005, [0007].

The information collected from the sensors represents, for instance, “audio, pressure, or temperature” (*environment information*). Ex.1005, [0010]. Ex.1003, ¶77.

Thus, Jakobson discloses or at least renders obvious the *sensor network* sensing *environment information*, alone or in view of Tsetsos. Ex.1003, ¶¶74-78.

[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*

As discussed above in [1.1], Jakobson’s FIG. 7 illustrates that its middleware 25 receives sensor data from “sensor networks” (blue) and transmits the sensor data to the signal and data fusion services via the “Fast Real-Time Event Channel 29.” Ex.1005, [0216], [0219].



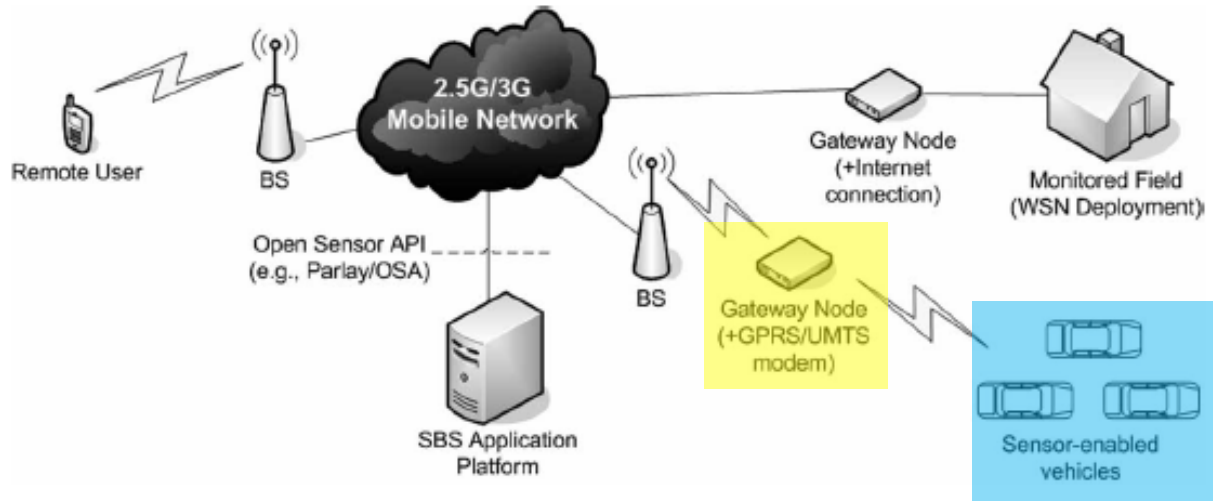
Ex.1005, FIG. 7 (partial annotated); Ex.1003, ¶79.

To the extent Jakobson doesn't explicitly disclose that the sensor networks transmit data to the middleware via a "*sensor network data transmitter*," doing so was well known and thus would have been obvious. Ex.1003, ¶¶32-33.

For instance, Tsetsos describes that typical modern sensor networks utilize a "gateway node"³ (*sensor network data transmitter*), which operates as the

³ The Background of the '232 patent acknowledges that gateways for sensor networks were known. Ex.1001, 1:50; Ex.1003, ¶¶32-33.

“connecting link between the sensor network and the WAN (i.e., mobile network or internet).” Ex.1006, 3. In FIG. 1, Tsetsos illustrates how the Gateway Node (yellow) links a domain of sensor-enabled vehicles (blue) to a mobile network.



Ex.1006, FIG. 1 (annotated); Ex.1003, ¶81.

In one implementation, Tsetsos explains that a Gateway node operating under an Open Services Gateway Initiative (OSGi) protocol, which has been “adopted by major industries,” aggregates sensor data and provides the aggregated sensor data in a standardized format. Ex.1006, 7.

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).” Ex.1006, 3; Ex.1003, ¶¶81-83.

It would have been thus obvious for Jakobson's domain of sensors to communicate through a gateway like the one in Tsetsos's to leverage the increased computational power and communication capabilities. Ex.1003, ¶84. 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. *See* Ex.1006, 3 ("typical modern sensor network" includes a "gateway node"), 5 (describing Open Services Gateway Initiative (OSGi)); Ex.1003, ¶¶84-86, 32-33. Using a sensor gateway would have reduced the hardware and software requirements for the individual sensors, and allowed for efficient communication of aggregated sensor data from the sensor nodes to the middleware services. *See* Ex.1006, 3 (gateways have more communication capabilities than sensors), 7 (sensors have limited processing power); Ex.1003, ¶¶85-86, 32-33.

A POSITA would have recognized the advantages of funneling heterogenous sensor data through a gateway designed for that purpose, instead of managing several different sensor communication pathways. Ex.1003, ¶85. For instance, a gateway could simplify the process of adding or removing sensors from the sensor domain. *See* Ex.1006, 2 (wireless sensor networks frequently change and adapt); Ex.1003, ¶85. This provides benefits analogous to those provided by Jakobson's protocol conversion functions. *See* Ex.1005, [0014]-[0015] (event

collection permits “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”); Ex.1003, ¶85. Indeed, a POSITA would have understood that the object of middleware is to facilitate communication and translation between heterogeneous system components that use different protocols and data formats. *See* Ex.1007, 526 (“middleware serves as a translation mechanism, gluing together application software across a network”); Ex.1006, 5 (middleware provides “unified handling” of heterogeneous networks), 6 (middleware supports “ubiquitous context-aware services”); Ex.1003, ¶85.

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. Ex.1003, ¶86.

In the combination, the gateway would *collect*[] the sensor data and *transmit*[] *the collected sensed data*, as described in Tsetsos. Ex.1003, ¶87.

Second, data from Jakobson’s sensors is “real-time data.” *See* Ex.1005, [0221], FIG. 8 (“**Real Time** Data Feed”). Thus, the data collected from the sensors

is collected and provided *in real-time*. See also, Ex.1005, [0216], [0219]. As explained above, it would have been obvious to utilize Tsetsos's gateway to collect the sensor data from various sensors and forward the data to Jakobson's SBM middleware. 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. Ex.1003, ¶88.

Third, with respect to “*transmitting...in a message form*,” Jakobson teaches that the SBM middleware receives the sensor data as “events.” Ex.1005, [0011] (“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”), [0175] (events are “provided from the domain’s instrumentation....”).

Jakobson refers to an event 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.” Ex.1005, [0011]-[0012]; Ex.1003, ¶¶89-91 (citing Ex.1007, 517 (a “message” is a “sequence of characters used to convey information or data”)). Similarly, Jakobson explains that its event collection and event distribution mechanisms facilitate processing of “messages.” Ex.1005, [0014]-[0016].

Thus, Jakobson discloses that the sensor networks provide an event *in a message form*. Ex.1003, ¶92.

Additionally, Tsetsos explains that its gateway aggregates the sensor network data and transmits the data via the “internet.” Ex.1006, 3, 7. Data transmitted via the internet is sent in IP packet form (“*message form*”). Ex.1003, ¶92; Ex.1007, 812-813 (“TCP/IP” transmits “messages” as “packets”), 437 (“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); Ex.1008, 41-44 (describing TCP/IP packet handling procedures); *see also*, Ex.1001, 4:35-37 (the gateway device collects the data from the various sensors and transmits the data using “Transmission Control Protocol/Internet Protocol (TCP/IP)”).

Tsetsos also describes using standardized “open public [Application Programming Interfaces (APIs)] like “Parlay/OSA (Open Services Access)” to facilitate “sensor data retrieval.” Ex.1006, 3. It would have thus 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* (e.g., IP packets, API requests), consistent with Jakobson’s approach for handling events provided by the instrumented domain. Ex.1003, ¶92.

A POSITA would have found it obvious to use a standard message format

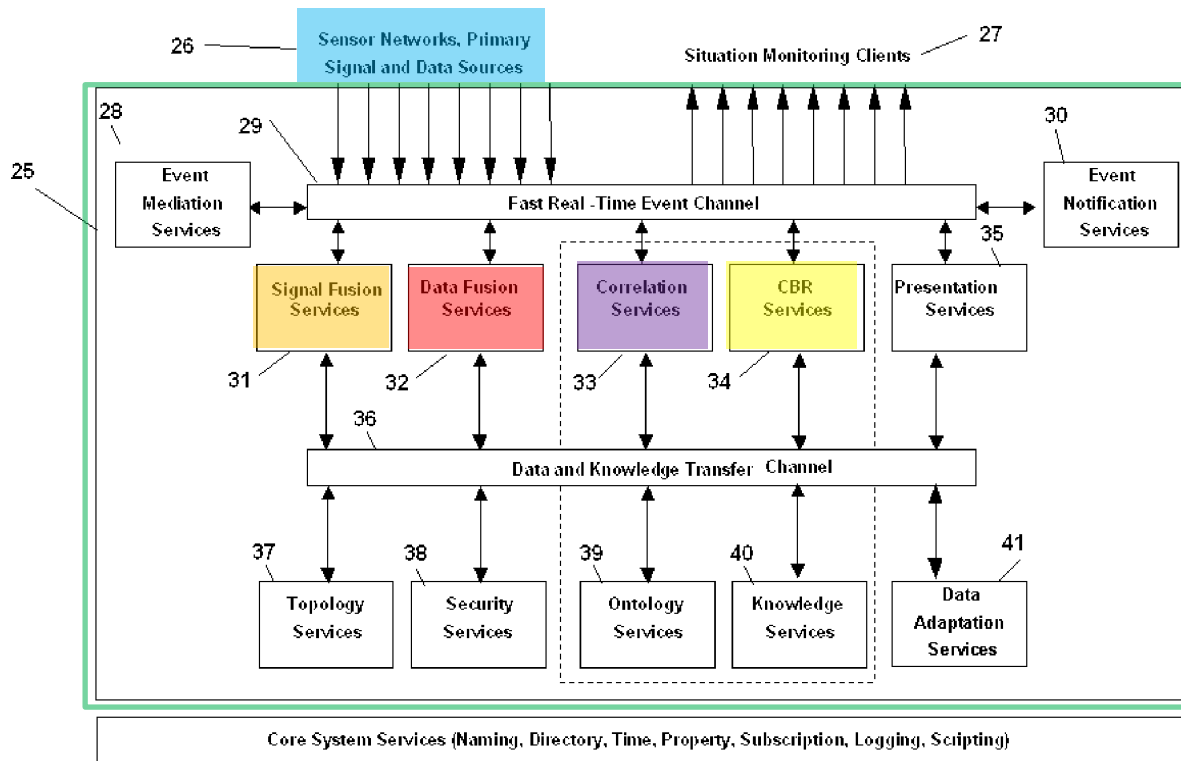
for transmitting the aggregated data (e.g., as IP packets) to improve the reliability and robustness of the data communication. Ex.1003, ¶¶92-93. For instance, a POSITA would have relied on well-established and developed digital communication protocols (e.g., the Internet Protocol (IP), 3GPP protocols, 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. Ex.1003, ¶¶34-35, 94-95.

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* Ex.1005, [0280] (operations “realized on a multiprocessor computer”); [0215] (SBM middleware realized by software architecture); *see also* Ex.1006, 3 (gateway nodes with processing and communication capabilities), 5 (mediation software based on Open Services Gateway Initiative); Ex.1003, ¶97.

Thus, Jakobson discloses or renders obvious this limitation, alone or in view of Tsetsos. Ex.1003, ¶97.

[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,

First, Jakobson discloses “middleware services” 25 (green box) for realizing situation based management (SBM). Ex.1005, [0216].



Ex.1005, FIG. 7 (annotated); Ex.1003, ¶98.

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).

Jakobson's middleware services 25⁴ renders obvious the “*Ubiquitous Sensor Network (USN) middleware*.” Ex.1003, ¶¶98-101.

Jakobson explains that data fusion (red) operations of the event collection services include “filtering, translating, cleansing, aliasing, aggregating, and converting to a uniform representation.” Ex.1005, [0013]. Jakobson's event correlation service “**extracts the essential properties of the events** collected by event collection 3.” Ex.1005, [0175]. Additionally, the information output to the situation viewer 14 includes “**extracts** of situations.” Ex.1005, [0178]-[0179]; Ex.1003, ¶102.

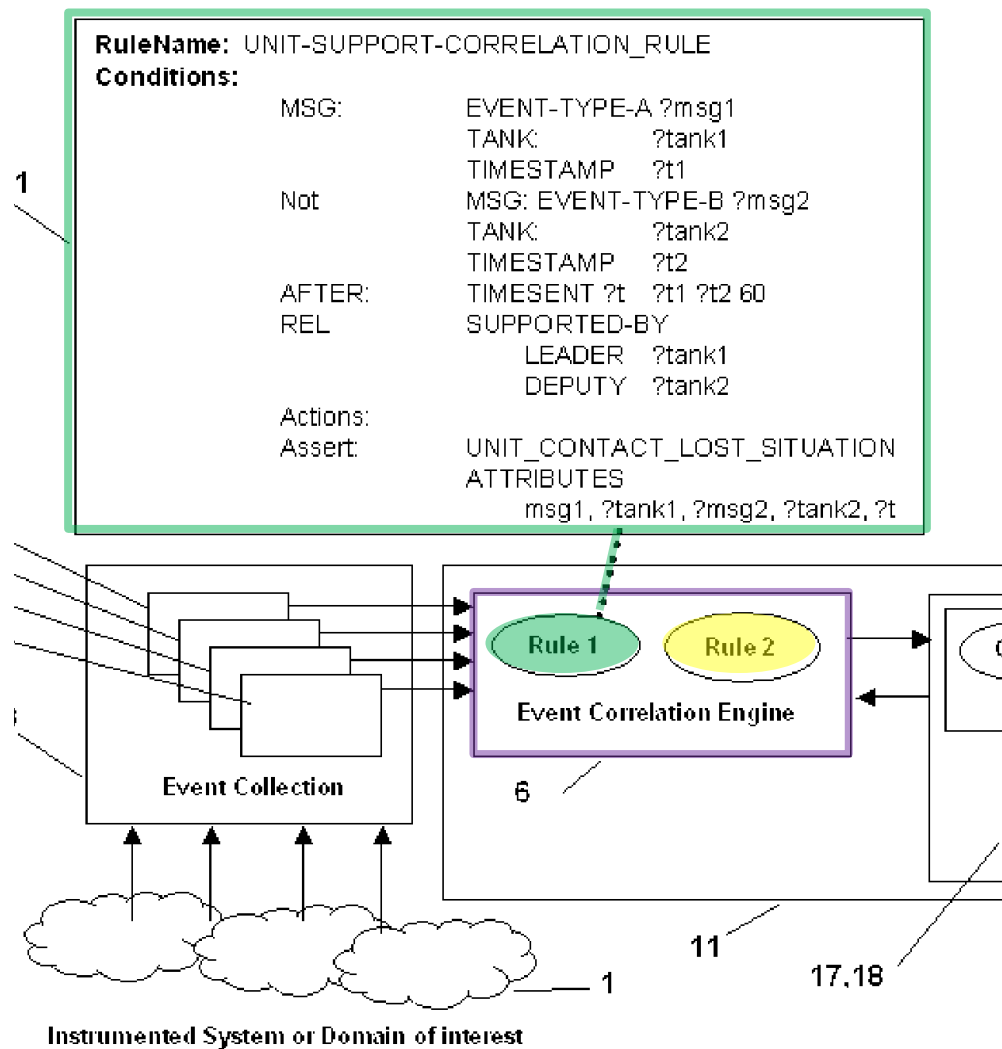
Thus, Jakobson discloses that the middleware 25 (*USN middleware*) *extract[s] object and environment information*. Ex.1003, ¶103.

Second, with respect to “*cleaning, classifying, and integrating*,” referring to

⁴ In this Petition, the colored boxes encapsulating the various modules mapped to the claim elements (e.g., the *USN middleware*) do not necessarily indicate that every module within the box should be mapped to the corresponding element, or that the claim element at issue cannot include components or functionalities not expressly shown within the box.

FIG. 6⁵, Jakobson explains that the data fusion module performs filtering, cleaning, aggregating, and format conversion operations to provide events ready for further processing by the event correlation module. Ex.1005, [0013], [0182]. The event correlation module uses rules engines to execute various correlation rules (green and yellow below), such as event “compression,” “filtering,” and “counting.” Ex.1005, [0185]-[0187]. Ex.1003, ¶104.

⁵ As explained above with respect to limitation [1.0], Jakobson illustrates its SBM systems using various implementation examples, including the distributed services software framework of FIG. 7. However, the functionalities in each implementation example are the same (e.g., signal and data fusion, event correlation, situation management (CBR)).



Ex.1005, FIG. 6 (partial annotated); Ex.1003, ¶104.

As explained below, these steps performed by the data fusion module, the event correlation module, or both, disclose or render obvious the *cleaning*, *classifying*, and *integrating* steps performed by the *USN middleware* of limitation [1.3]. Ex.1003, ¶¶105-106.

“cleaning”

In a **first example**, Jakobson’s data fusion operations include “filtering” and “cleansing” the data provided from the sensors. Ex.1005, [0013], [0182]. Filtering or cleansing the data discloses “*cleaning...the sensed data.*”

In a **second example**, the event correlator’s event filtering operation involves discarding events if their associated “priority, type, location, [or] time stamp...does not fall into the set of predefined legitimate values.” Ex.1005, [0187]. Event filtering therefore discloses or renders obvious “*cleaning*” the events (*sensed data in the message*), because it discards events that do not meet certain criteria. Ex.1003, ¶¶107-109.

“classifying”

In a **first example**, Jakobson’s data fusion operations include “**data pattern recognition**” and “tracking object movements.” Ex.1005, [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. Ex.1003, ¶110.

In a **second example**, Jakobson’s event correlation operations include “Event escalation” which “assigns a higher value to some parameter” of the event, such as “**severity**” depending on “the operational context, e.g., the number of occurrences of the event.” Ex.1005, [0188]. The event escalation including priority

assignment is an example of *classifying* the sensed data, because this operation involves sorting and grouping events (e.g., by severity) based on a parameter (e.g., number of occurrences). Ex.1003, ¶¶111-112.

In a **third example**, Jakobson’s event correlation operations also include event “generalization,” which categorizes an event into “a super class,” and event “specialization,” which categorizes an event into “a more specific subclass.” Ex.1005, [0188]. Event generalization and specialization are also examples of *classifying the sensed data*, because they assign (*classify*[]) events to a “super class” or a “subclass,” respectively. *See also*, Ex.1005, claim 19 (“apparatus” comprising “means to **classify** information”). Ex.1003, ¶¶113-115.

Thus, Jakobson discloses or renders obvious the data fusion module, the event correlation module, or both, *classifying* the sensed data.

“integrating the message”

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. Ex.1005, [0182], [0013], [0221]. Aggregating the data to prepare the data for event correlation discloses or renders obvious “*integrating*” the data. Ex.1003, ¶117.

In a **second example**, Jakobson explains that the event correlation combines a “collection of events” (*integrating the message*) to infer new events. Ex.1005,

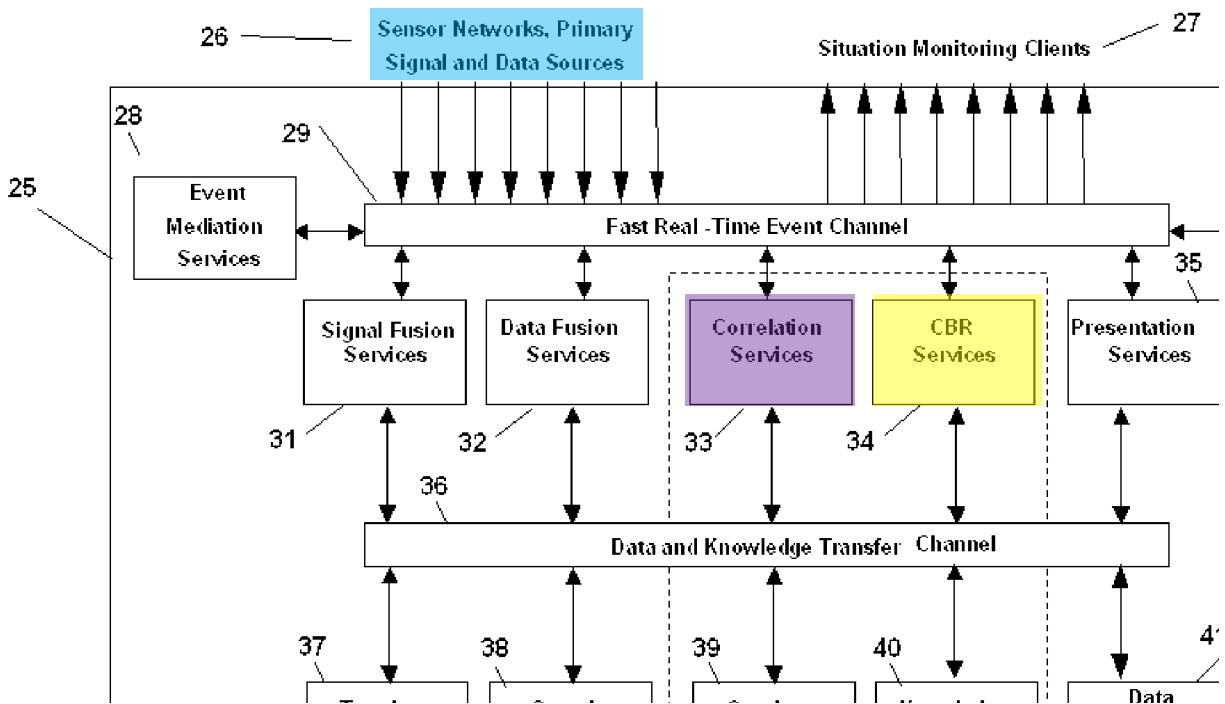
[0307]. For instance, Jakobson’s event correlation operations “reduce a collection of events to a root-cause or a high-level inferred synthetic event.” Ex.1005, [0019]. Combining and correlating lower-level events discloses or renders obvious *integrating the message*. Ex.1003, ¶¶118-120.

Third, as explained above, Jakobson discloses that the result of the event correlation actions, including those discussed above, is to “**extract[] the essential properties of the events** collected by event collection 3.” Ex.1005, [0175]. Thus, Jakobson discloses or at least renders obvious that the data fusion operations, the event correlation operations, or both *extract[] object and environment information by cleaning, classifying and integrating the events (the message[s])* provided from the sensor networks. Ex.1003, ¶¶121-122.

Thus, Jakobson renders obvious limitation [1.3].

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

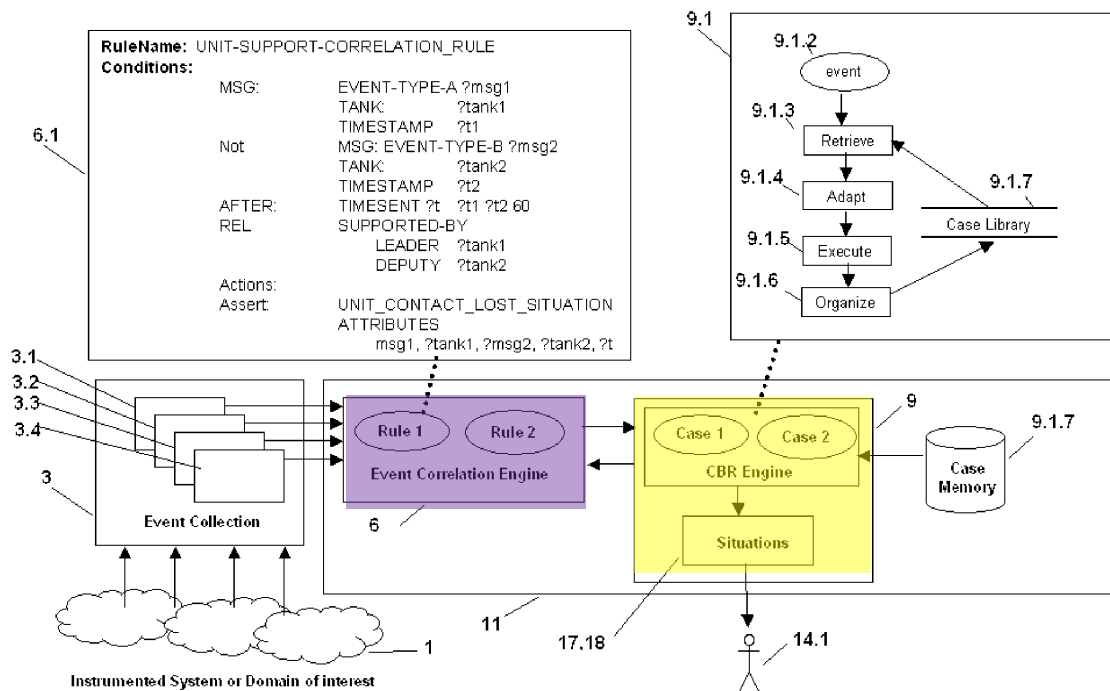
Jakobson’s SBM middleware (*USN middleware*) shown in FIG. 7 includes correlation services (purple) for performing event correlation tasks like filtering, compression, and escalation. Ex.1005, [0217], [0219]. The SBM middleware also includes case-based reasoning (CBR) services (yellow) for situation management tasks, including creating and modifying situations, and providing contextual information associated with those situations. Ex.1005, [0217], [0219].



Ex.1005, FIG. 7 (partial annotated); Ex.1003, ¶123.

Similarly, Jakobson discloses that the SBM 11 includes an event correlation module (purple) and a situation manager (yellow) that includes a CBR engine.

Ex.1003, ¶¶123-124.



Ex.1005, FIG. 6 (annotated); Ex.1003, ¶124.

“generating conditional events”

Jakobson’s event correlation module generates several types of higher-level “inferred” events (*conditional events*) from the low-level sensor data. See Ex.1005, FIG. 7, [0019], “Correlation Services.” Jakobson uses the term “event” to refer to a range of lower-level and higher-level information, from raw sensor data ([0181]) to complex patterns ([0188]), and “root-cause” synthetic events ([0019]) identified based on the sensor data. Ex.1003, ¶125.

Jakobson describes generating high-level inferred events based on “Boolean operators over **conditional** (predicate) terms.” Ex.1005, [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.” Ex.1005, [0175]. A POSITA would have understood that the higher-level inferred events generated based on these conditional tests and predicates are *conditional*. Ex.1003, ¶¶126-127.

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.”**

Ex.1005, [0176]; Ex.1003, ¶¶128-129.

In another thresholding example, event correlation includes “counting and thresholding the number of repeated arrivals of identical events.” Ex.1005, [0188]. Jakobson’s threshold-based events are *conditional* events because they are conditioned on the comparison to a threshold. Ex.1003, ¶¶127, 130-131.

Jakobson’s higher level inferred events are consistent with the ’232 patent’s description and examples of “conditional events.” *See* ’232 patent, Table 1 (“low temperature: 13.5...for average temperature value”); Ex.1003, ¶129.

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.

Ex.1001, Table 1 (partial annotated); Ex.1003, ¶129.

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.*” Ex.1003, ¶131.

Accordingly, Jakobson’s event correlator generating high-level inferred events discloses or renders obvious *generating conditional events.* Ex.1003, ¶132.

“context aware information”

Jakobson describes multiple examples of “*generating...context aware information.*”

In a **first example**, Jakobson’s situation management module uses case-based reasoning (CBR) services to generate “situations” from the correlated events, which are based on the low-level sensor data. *See* Ex.1005, FIG. 7, [0219].

Jakobson generates “situations” (*context aware information*) when a collection of events from the event correlation module together meets the criteria of a pre-defined “situation template.” Ex.1005, [0052], [0026]. For instance, Jakobson explains that a situation represents a high-level state, such as “power disruption” or “chemical release.” Ex.1005, [0177]. Ex.1003, ¶¶133-134.

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

Ex.1005, Fig. 3 (partial annotated); Ex.1003, ¶134.

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.

Ex.1001, Table 1 (partial annotated); Ex.1003, ¶135.

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.

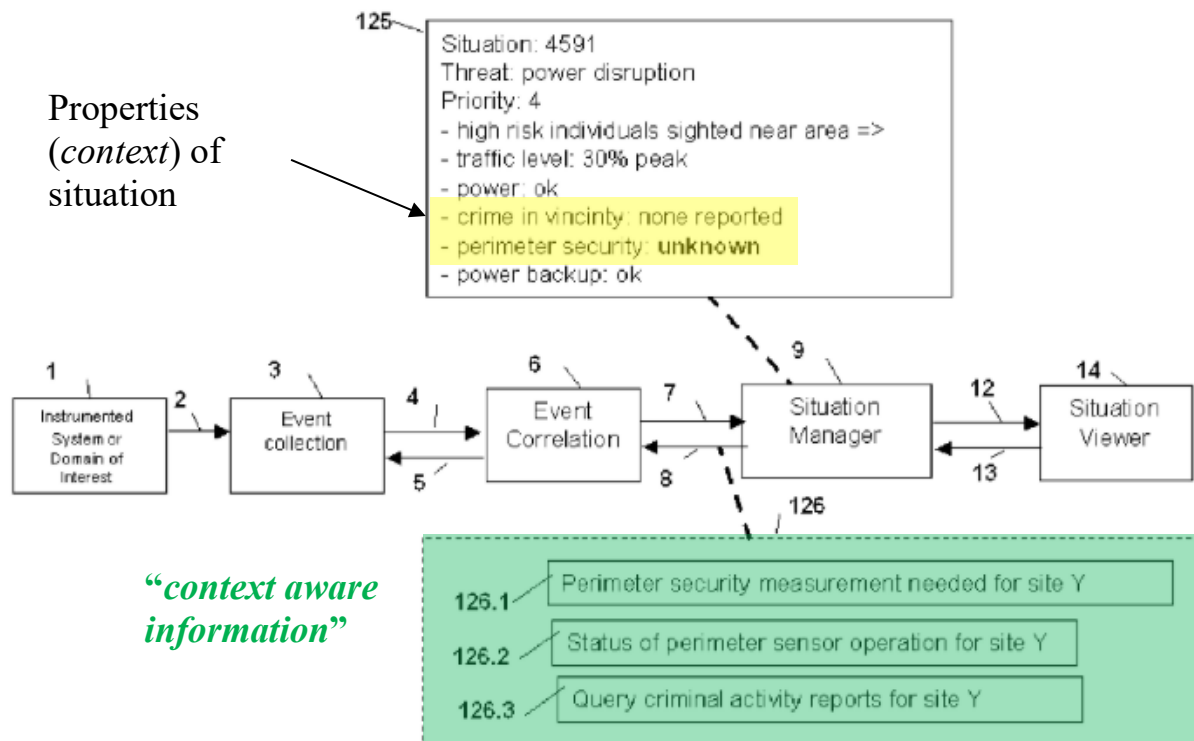
Ex.1001, Listing 1; Ex.1003, ¶136.

In a **second example**, Jakobson's situation manager also generates the "associated **context**" for a situation. Ex.1005, [0027]-[0029]. Jakobson refers to this context as "**context-sensitive information**" (*context aware information*), which the situation manager provides to the event correlator. Ex.1005, [0282], [0307]. Ex.1003, ¶137.

Jakobson's "context-sensitive information" for a situation may relate to assessing or improving the confidence level of a situation. For instance, Jakobson explains that the "degree of confidence" of a situation may be "less than perfect." Ex.1005, Abstract, [0282]; *see also* [0298], [0322] (determining confidence levels of situations). Jakobson's SBM may "take action" to provide instructions (*context aware information*) to the event correlator to request information that would increase the confidence level in the situation (e.g., missing parameter values). Ex.1005, [0282]. Ex.1003, ¶138.

Referring to FIG. 18 below, Jakobson describes the situation manager feeding back operational context information (*context aware information*) 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 [sic]." Ex.1005, [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 (green) are messages or requests that are created and provided to the event correlator based on the properties the situation manager identifies. Ex.1003, ¶139.



Ex.1005, FIG. 18 (annotated); Ex.1003, ¶139.

Thus, the information 126.1, 126.2, 126.3 is *context aware information* because it is “**based on the context** provided by the current situation.” Ex.1005, [0299]; *see also*, FIGS. 19 and 20 (additional *context-aware information* examples, including withdrawal of query, indication of inconsistent sensor readings, and indication of inferred impersonation attempt). Ex.1003, ¶140.

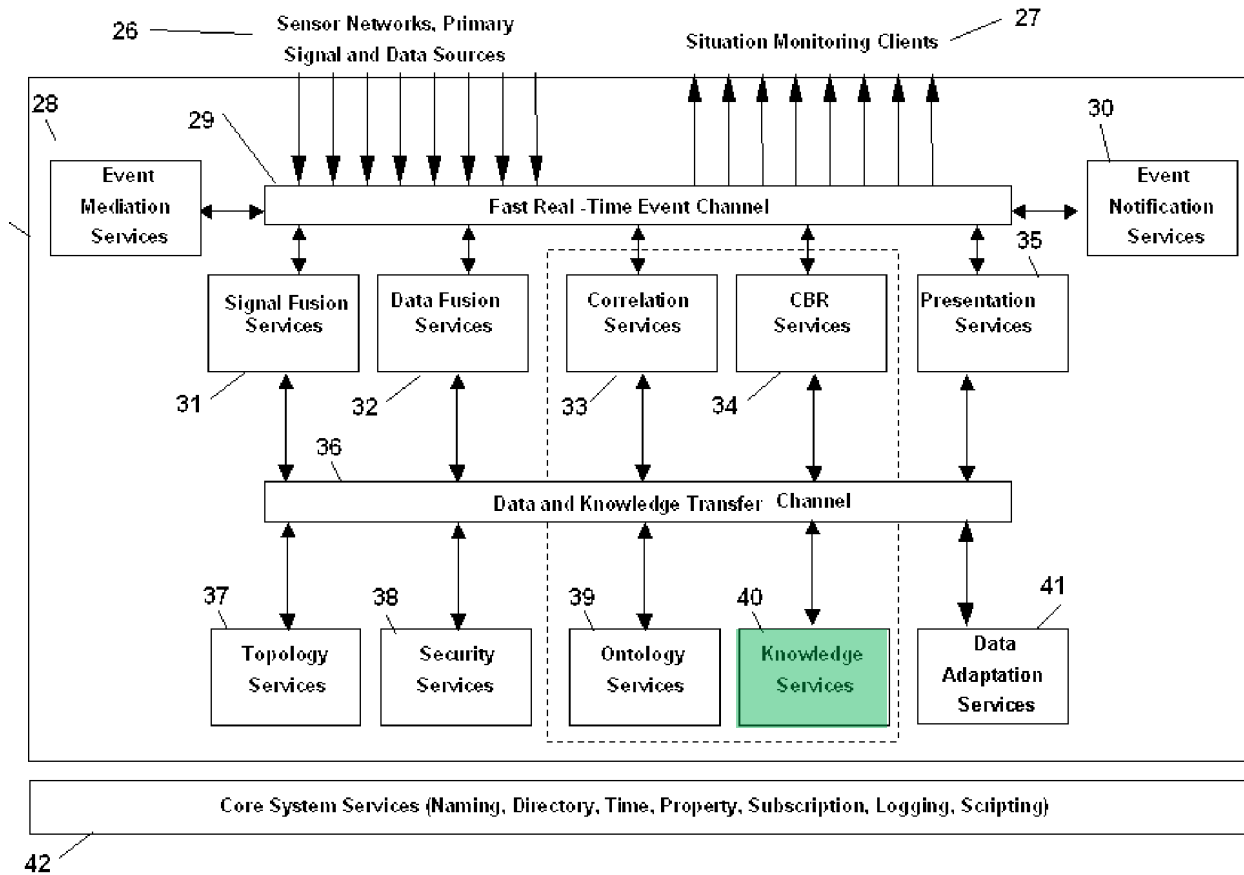
In a **third example**, Jakobson describes providing ranking values for

different situations, depending on what types of parameters are available (e.g., parameter x or parameter y). Ex.1005, [0296]. Based on information about which parameters are available, “alternative situations can be ranked with certainty factors based on the available information.” Ex.1005, [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.” Ex.1005, [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*. Ex.1003, ¶141.

Accordingly, Jakobson’s situations, context-sensitive information, or situation rankings disclose or render obvious *generating...context aware information*. Ex.1003, ¶¶133-141.

“knowledge contents”

Referring to Fig. 7, Jakobson explains that the SBM middleware includes a “knowledge service 40,” (green). Ex.1005, [0217], [0219].

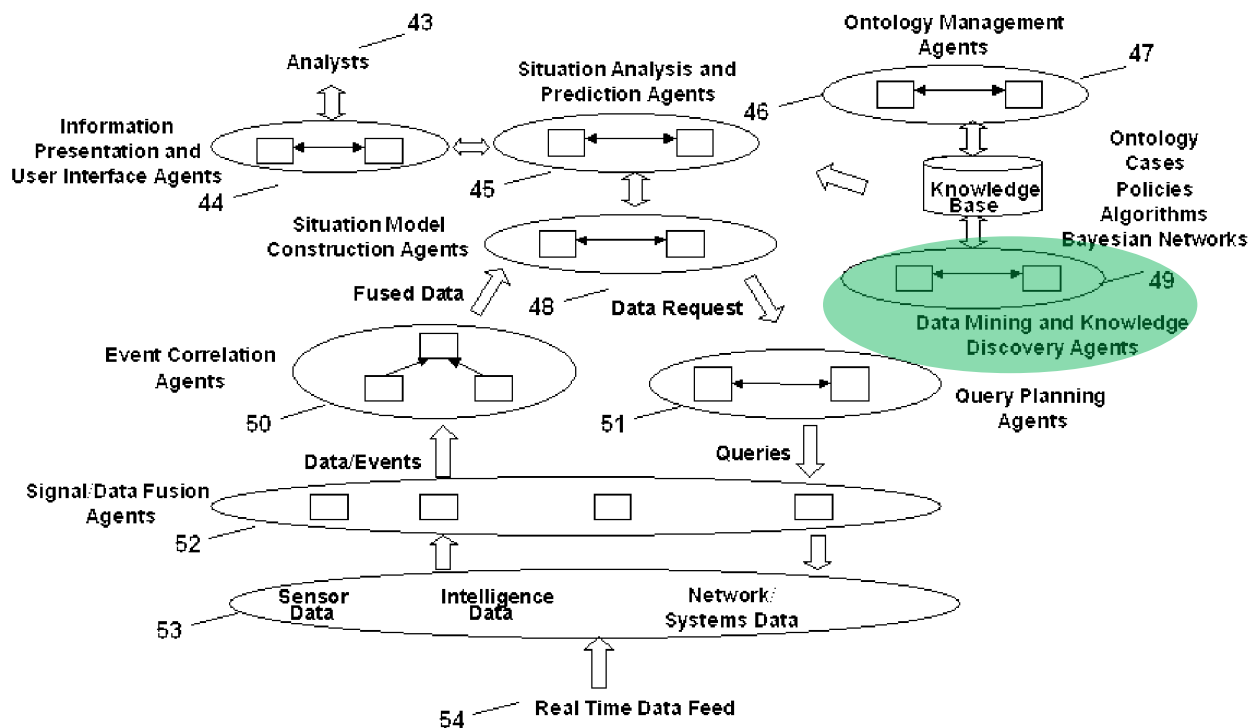


Ex.1005, Fig. 7 (annotated); Ex.1003, ¶142.

The knowledge services provide “domain modeling, knowledge representation, and reasoning facilities used by the Event Correlation Service 33 and the Case-Based Reasoning Service 34.” Ex.1005, [0217]. Ex.1003, ¶¶142-143.

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.” Ex.1005, [0223].



Ex.1005, Fig. 8 (annotated); Ex.1003, ¶144.

The information contained in the knowledge services, ontology services, or both, includes *knowledge contents*. Ex.1003, ¶145. Jakobson explains that the knowledge base and ontology services can be modified by the situation viewer. Ex.1005, [0180].

Thus, Jakobson discloses modifying *knowledge contents* stored in the knowledge base. 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 accuracy of situation management, and to provide more reliable information. Ex.1003, ¶¶147-148.

As another example shown in FIG. 3, Jakobson's SBM system generates a set of properties (*knowledge contents*) (green) for 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

Ex.1005, FIG. 3 (partial annotated); Ex.1003, ¶149.

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

Ex.1005, [0309]-[0310], [0114]-[0115]. Ex.1003, ¶¶149-150.

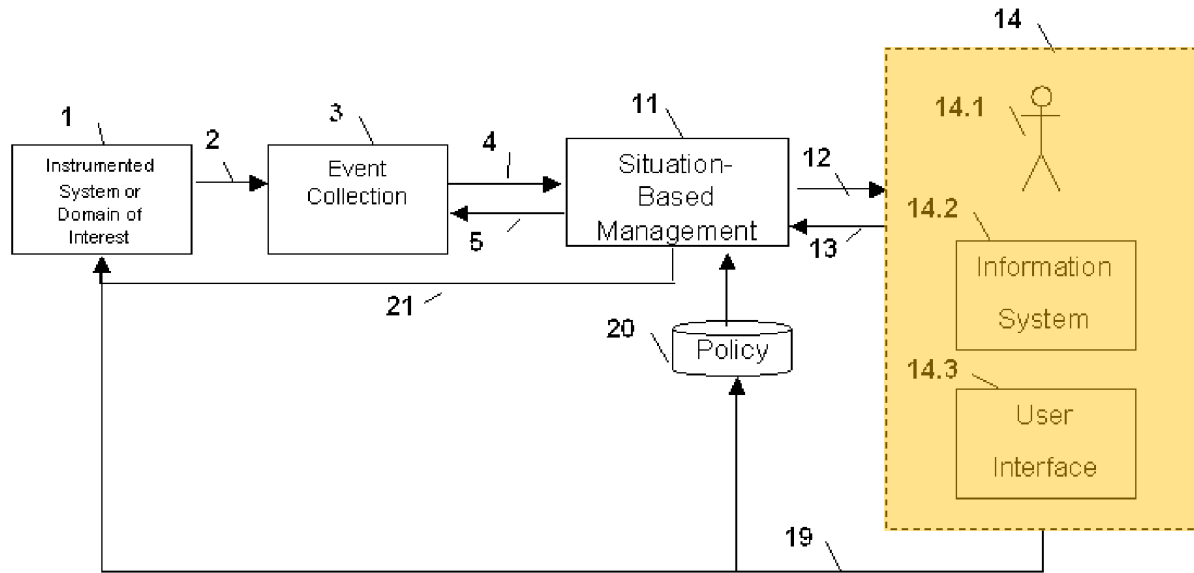
Accordingly, in addition to the “knowledge bases,” Jakobson’s situation properties and situation transition graphs are *knowledge contents*.

Thus, Jakobson discloses or renders obvious this limitation. Ex.1003, ¶¶151-152.

[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.*

Jakobson’s SBM system includes a “situation viewer” that provides “situation reports,” “events,” and other situation management information (*the information service*) to an “information system,” or “a user interface” (*an application program*). Ex.1005, [0178].

FIG. 14 illustrates the situation viewer (orange), which can include an information system 14.2 and/or a user interface 14.3. Ex.1003, ¶¶153-154.



Ex.1005, FIG. 4 (annotated); Ex.1003, ¶154.

Jakobson explains that the “situation reports” and other situation information (*the information service*) are “presented” to a user by the user interface, that the user can “control the presentation” using a keyboard or pointing device, for example. Ex.1005, [0179]. Jakobson also explains that such information can be “provided” to a user “through an instant message, an email, and a voice, image or video input system.” Ex.1005, [0178]; Ex.1003, ¶¶154-157.

Because this situation-related information is based on the sensor data (events) provided by the sensor network of the domain (Ex.1005, [0168]-[0170], [0181]), the information is *related to environment where the sensor network is installed*. Ex.1003, ¶158.

Accordingly, providing situation reports or other situation-related

information to the user via a user interface, to an information system, and/or through communication applications (e.g., instant message, email) discloses “*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.*” Ex.1003, ¶¶157 (obvious for user interface and/or information system to include an application program), 159.

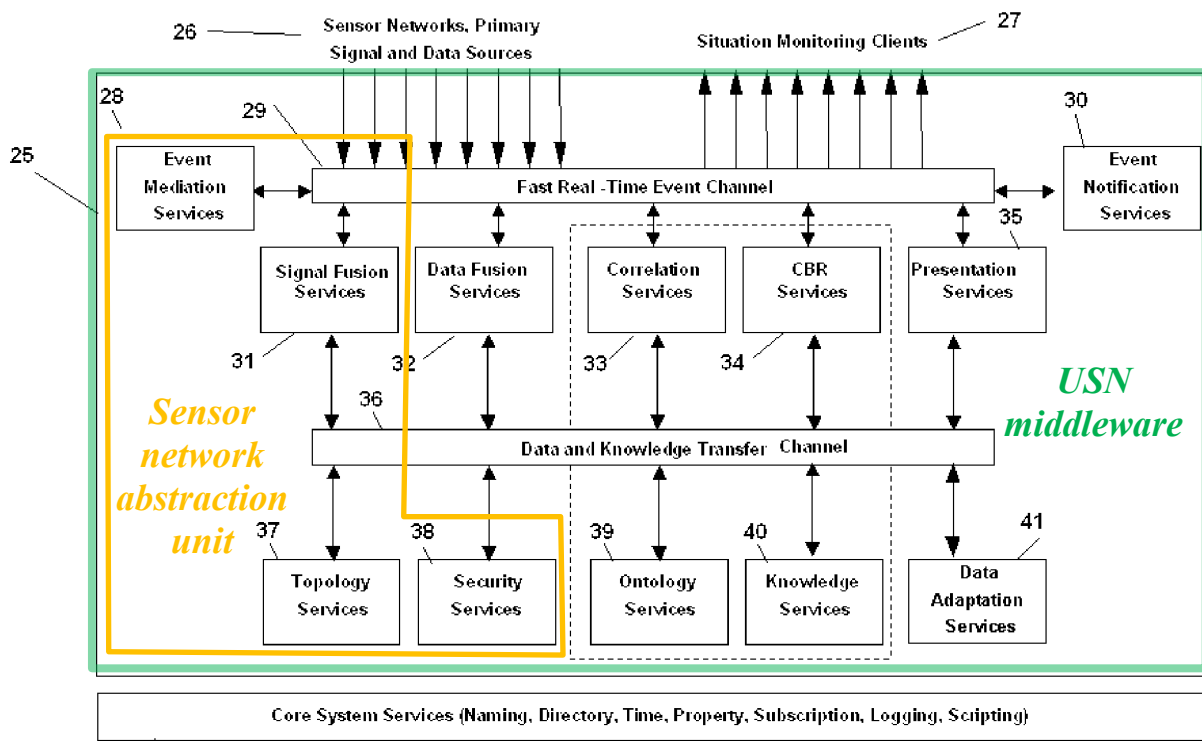
4. Claim 2

[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;*

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. Ex.1005, [0012].

Jakobson’s SBM system includes a “signal fusion” module that “extract[s] information” from the sensor signals (*sensed data*). Ex.1005, [0010]. For instance, the objects of signal fusion include “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.” Ex.1005, [0010]. Thus, the signal fusion module *extract[s] the sensed data from the message*. Ex.1003, ¶¶160-161.

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 “transports events between services,” using “connectivity and protocol conversion functions” to provide the sensor data to the SBM middleware. Ex.1005, [0216].



Ex.1005, FIG. 7 (annotated); Ex.1003, ¶162.

Accordingly, event channel and event mediation services provide an interface between the sensor networks and the signal fusion module, which *extract[] the sensed data* from the events received from *the sensor network*. Ex.1003, ¶¶162-163.

Thus, Jakobson's signal fusion module, event mediation module, and the event channel⁶, 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*. Ex.1003, ¶164.

[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

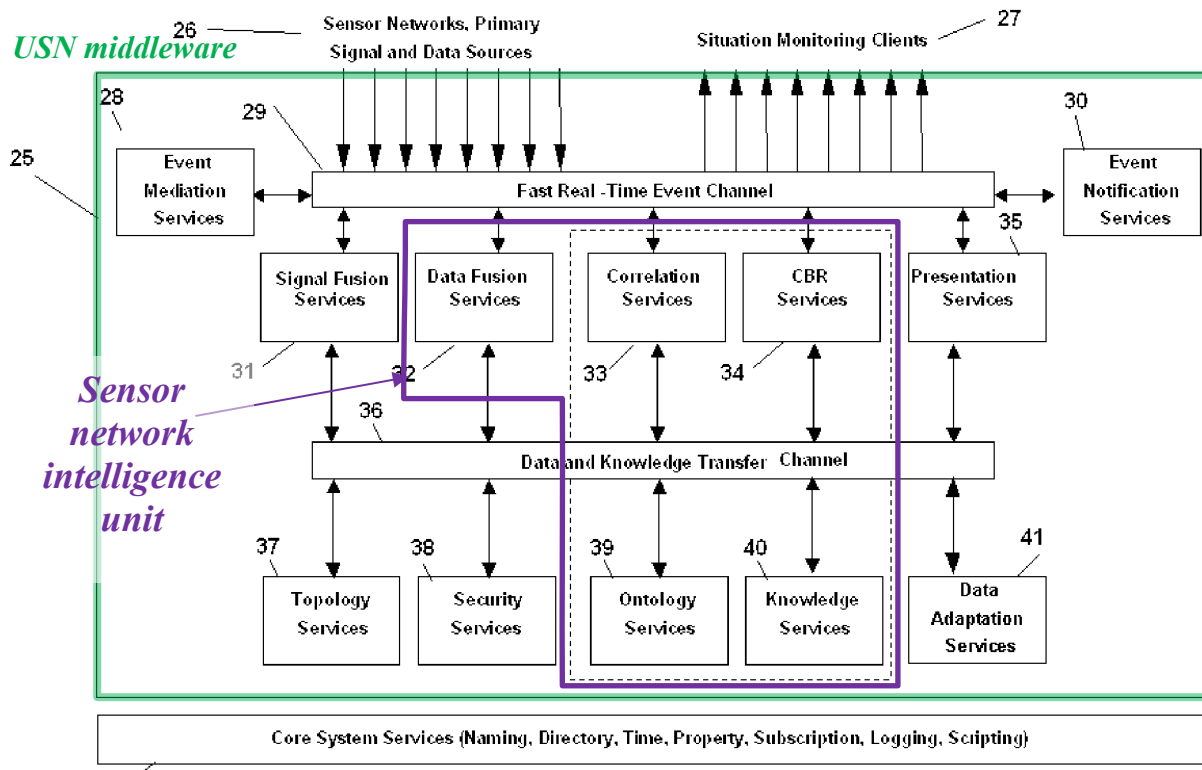
As explained above for limitation [2.0], the signal fusion module, event mediation, and event channel (*sensor network abstraction unit*) *extract[] the sensed data*. Jakobson explains that the sensed data extracted by signal fusion is provided to the data fusion module, and then onto the event correlation. Ex.1005, [0013]; Ex.1003, ¶167.

As explained above for limitation [1.3], the data fusion module, the event correlation module, or both, extract the *object and environment information by analyzing the sensed data*.

As also explained above for limitation [1.4], the event correlation module

⁶ As explained with respect to limitation [3.2] below, the *sensor network abstraction unit* may also include the topology services module 37 and the security services module 38.

and the situation manager (CBR Services), disclose *generating the conditional events* and the *context aware information*. Ex.1003, ¶¶165-166



Ex.1005, FIG. 7 (annotated); Ex.1003, ¶166.

Thus, Jakobson's data fusion services, correlation services, and situation manager (CBR services) disclose the *sensor network intelligence unit*. 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*). Ex.1005, [0013] (events from signal fusion are provided to data fusion and then to event correlation). Ex.1003,

¶¶167-168.

“extracting the object and environment information”

Jakobson explains that the event correlation module “**extracts** the essential **properties** of the events” provided from the sensor networks via the event collection module. Ex.1005, [0175].

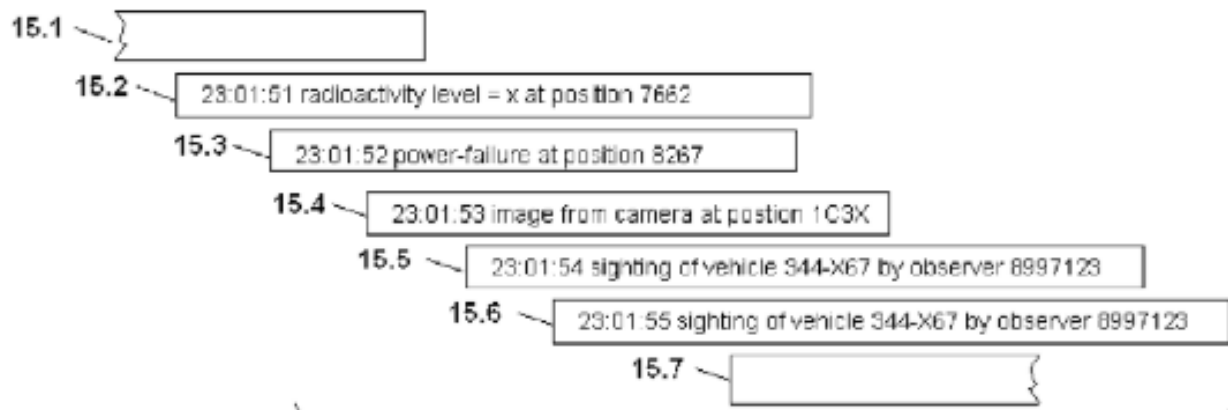
The extracted “properties” include *object and environment information* because the properties represent a condition of the domain and are based on the sensed environment data from the domain. Ex.1005, [0170]-[0173], FIG. 3. Thus, Jakobson discloses that the data fusion services, event correlation module, or both (corresponding to *the sensor network intelligence unit*) *extract[] the object and environment information*. Ex.1003, ¶¶169-170.

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

First, as explained above at limitation [1.4], event correlation operations to generate high-level inferred events discloses “*generating the conditional events.*” Ex.1003, ¶172.

Second, Jakobson explains that the inferred high-level events (*conditional events*) include a “*notification and alarm about a certain condition.*” For instance, Jakobson refers to the high-level inferred events as “notification[s],” and explains that events can include “human-readable text.” Ex.1005, [0012]. Ex.1003, ¶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*). Ex.1005, [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].”



Ex.1005, FIG. 2 (partial).

In a second example involving high-level inferred events, 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 vehicle). Ex.1005, [0191]. Ex.1003, ¶174.

A POSITA would have understood that each of these event messages produced by the event correlation includes a “*notification and alarm about a*

certain condition.” Ex.1003, ¶175. 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.” Ex.1001, 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. Ex.1003, ¶176. Thus, like the ’232 patent, Jakobson’s generation of high-level inferred events indicating a state of power failure, radioactivity, or lost contact discloses or renders obvious “*generating the conditional events including notification and alarm about a certain condition.*”

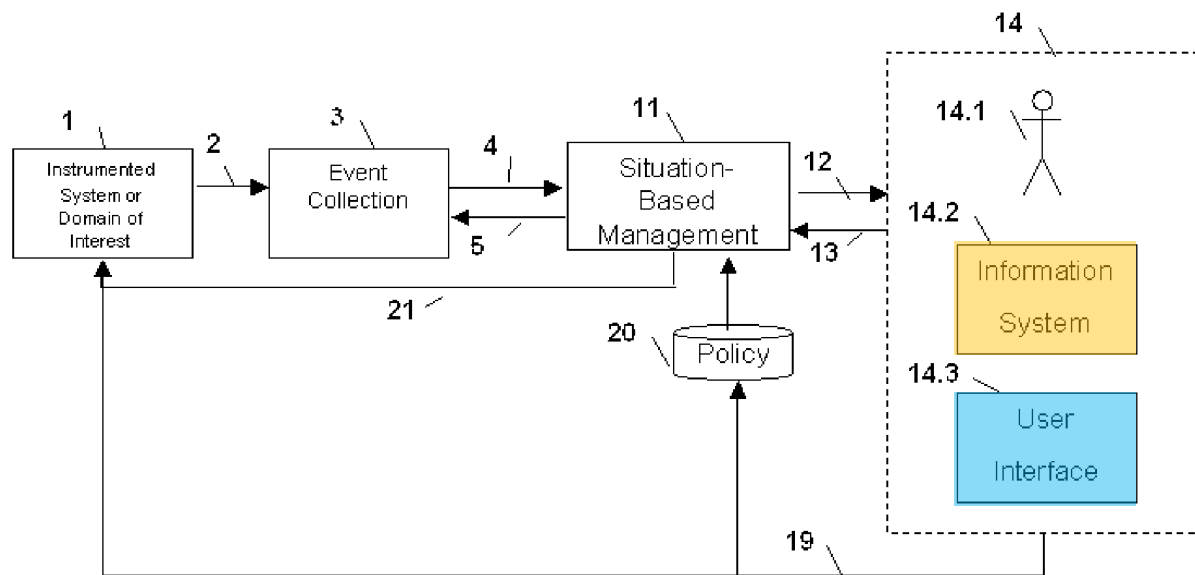
As explained above at limitation [1.4], Jakobson’s situation manager (part of *the sensor network intelligence unit*) generates situations (*the context aware information*) based on the data extracted and analyzed by the data fusion module and event correlation services (*from the extracted object and environment information*).

Thus, Jakobson renders this limitation obvious. Ex.1003, ¶¶177-178.

[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.*

Jakobson explains that the situation viewer 14 includes an “information

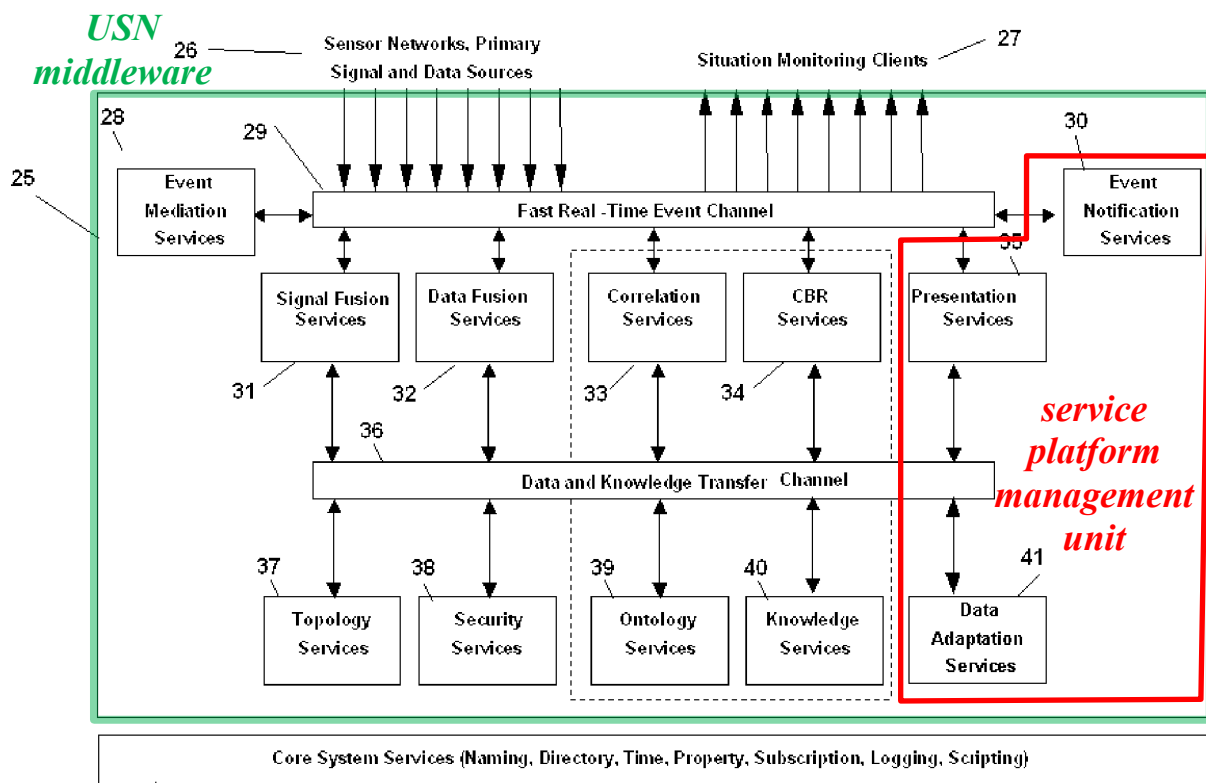
system 14.2” (orange) (*application program*) that receives reports of situations and situation-related information (*the context aware information*) that has been “converted to a different format” (*in a form suitable for the application program*). Ex.1005, [0178], [0179]. Jakobson also explains that situation-related information can be provided to a “user interface” (also an *application program*) (blue). Ex.1005, [0178], [0179].



Ex.1005, FIG. 4 (annotated); Ex.1003, ¶179.

Additionally, referring to FIG. 7, Jakobson’s middleware includes “Event Notification 30” which “provides facilities for subscribing to and filtering events.” Ex.1005, [0216]. The middleware 25 also includes presentation services 35 to “support the user interface” (shown in FIG. 4 above) and to present situations to “situation monitoring clients.” Ex.1005, [0218]-[0219]. Ex.1003, ¶¶179-180.

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 information system 14.2 and/or the user interface 14.3 (*in a form suitable for the application program*), and useful to help the viewer 14.1 understand and respond to situations.



Ex.1005, FIG. 7 (annotated); Ex.1003, ¶180.

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.

Ex.1005, [0179]; Ex.1003, ¶¶180-183.

Thus, Jakobson's event notification services and presentation services 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*. Ex.1003, ¶184.

5. 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;

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

Second, as explained above for limitation [1.2], Jakobson and Tsetsos render obvious *sensor network data transmitter* (gateway) transmitting messages to the SBM middleware in message 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). Ex.1003, ¶¶185-186; Ex.1007, 812-813 ("TCP/IP" transmits

“messages” as “packets”), 437 (“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); Ex.1008, 41-44 (TCP/IP packet handling procedures).

It would have therefore been obvious for the real-time event channel and/or the event mediation (*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*). Ex.1003, ¶¶187-188.

Additionally, Jakobson’s SBM techniques involve “**a protocol** for providing events from the [event collector] to the [situation manager].” Ex.1005, [0056], [0057]. Jakobson further explains that distributed system architectures use “standard inter-component **communication protocols**” (*message transfer protocol[s]*) to transfer information between system components. Ex.1005, [0215]. For instance, Jakobson discloses using software application frameworks like “CORBA” or “Java 2 Enterprise Edition.” Ex.1005, [0215]. These protocols use communication and application interface protocols, and are used as the “infrastructure of distributed systems” for system components encapsulated by software. Ex.1005, [0215]; Ex.1003, ¶¶189-190.

The communication protocols are used by Jakobsons’s “event mediation,”

which “performs the connectivity and **communication protocol conversion functions** so that sensor and intelligence data can reach” the fusion and event correlation services. Ex.1005, [0216]. Jakobson’s event channel 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*).

Accordingly, the event channel, the event mediation service, or both, disclose or render obvious the claimed *sensor network interface for extracting the sensed data by referring to a message transfer protocol*. Ex.1003, ¶¶191-192.

Additionally, it would have been obvious for the event mediation module to include, or function as, a *sensor network interface* in view of Tsetsos. Ex.1003, ¶¶193-195. 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). Ex.1006, 1-3. Tsetsos describes using communication protocols like “Parlay/OSA” and “OSGi” to handle communications between sensor devices and an application platform. Ex.1006, 3, 5. APIs based on those communication protocols can facilitate “the interaction between the applications and the network elements,” including “sensor data retrieval.” Ex.1006, 3. Ex.1003, ¶193.

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].” Ex.1006, 5. These APIs provide developers with functionalities for “user interaction, mobility management, user terminal capabilities, location, etc.” Ex.1006, 5; Ex.1003, ¶194.

It would have been obvious for the event mediation (*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. Ex.1003, ¶195.

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. Ex.1005, [0010]. Ex.1003, ¶¶196-197.

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*. Ex.1005, [0215]-[0216]; Ex.1003, ¶¶196-197.

Thus, the signal fusion services, alone or in view of Tsetsos’s standardized APIs and associated protocols, 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. Ex.1003, ¶¶198-208.

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

As explained above for limitation [3.0], Jakobson’s event mediation uses “connectivity and communication protocol conversion functions” to transfer data to the signal and data fusion services. Ex.1005, [0216]. The “connectivity and communication protocol conversion functions” performed by the event mediation disclose or render obvious *a rule for the message transfer protocols*. Ex.1003, ¶209.

For instance, it would have been obvious for the event mediation to include, or have access to, *storage* that stores instructions on how to extract data and manage the sensor networks according to networking protocols (e.g., TCP/IP) and/or using APIs (e.g., Parlay/OSA) to harmonize communications in a heterogeneous network. Ex.1003, ¶¶210, 34-35. 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 retrieve the payload for each type. *See* Ex.1005, [0007] (collecting and combining data from a variety of sensors using a variety of data processing networks); Ex.1006, 9 (challenges of

heterogenous wireless sensor networks). Ex.1003, ¶210. Indeed, Tsetsos explains that one of the very purposes of “middleware” is to facilitate unified handling of information from heterogenous sources (e.g., sensors). Ex.1006, 5, 8.

Thus, it would have been obvious for Jakobson’s event mediation, 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*. Ex.1003, ¶¶210-211, 34-35.

Additionally, Jakobson explains that the data fusion services convert data into “**a uniform representation.**” Ex.1005, [0013]. Because the data fusion services provides the data 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. Ex.1003, ¶¶212-213.

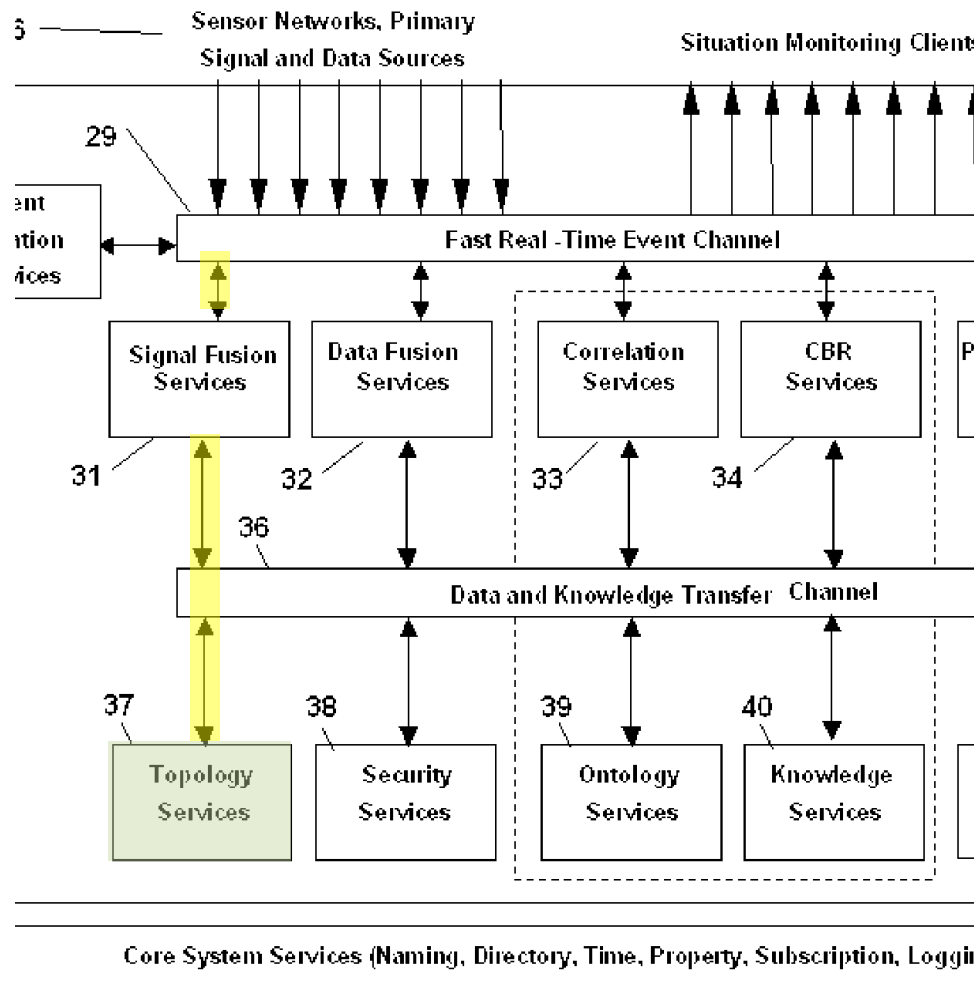
It would have been obvious for the data fusion services to include, or have access to, *storage* to store communication protocol rules for converting different types and forms of data to the “uniform representation.” Ex.1003, ¶214. Thus, the data fusion services also render obvious the claimed “*sensor network interface protocol storage for managing a rule for the message transfer protocol*” for similar reasons as explained above. Ex.1003, ¶214.

[3.2] a sensor network monitor for monitoring a state of the sensor network,

based on the data received from the sensor network interface.

Jakobson explains the SBM includes managing a domain having a collection of sensors, and that managing a domain includes “monitor[ing] and report[ing] the characteristics of the domain.” Ex.1005, [0009].

As one example of monitoring a domain, Jakobson’s SBM middleware includes a Topology module (*sensor network monitor*) to “store, represent, and manipulate information about connectivity, containment, and structural relationships.” Ex.1005, [0218]; Ex.1003, ¶¶215-217. As shown in FIG. 7 below, the topology module receives information from the event channel (*sensor network interface*) through the data and knowledge transfer channel, and the signal and data fusion services.



Ex.1005, FIG. 7 (annotated); Ex.1003, ¶218.

Topology information is “**continuously collected** from network elements,” and is used for “fault management” of networks to “**identify failures to operations technicians.**” Ex.1005, [0005]. In Jakobson’s SBM procedures, information about the “topology of a domain” of sensors is “combined with data processing models” and data collected from the sensors. Ex.1005, [0008]. Jakobson’s SBM methods include “determining and maintaining topological...and

other domain-specific relations between created situations.” Ex.1005, [0033]-[0035].

Thus, Jakobson discloses continuously collecting, from the sensors of the domain (*monitoring a state of the network*), topological information about the sensors. Ex.1003, ¶¶219-222.

To the extent Jakobson does not explicitly disclose that the topological information is collected by the Topology module *based on data received from the sensor network interface*, this would have been obvious. Ex.1003, ¶223. In that regard, as explained in limitation [3.0], the event channel, the event mediation , 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 through the event channel. It would have been obvious for the Topology module to rely on sensor data received and transmitted by the event channel to collect the topology information (e.g., message headers, sensor metadata, message routing information, hopcount, sensor status, etc.). Ex.1003, ¶223. This would have been an expected and efficient way to collect topology information, because it would not require dedicated communications between the sensor networks and the topology services. Ex.1003, ¶223.

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. Ex.1003, ¶224. For instance, Tsetsos explains that wireless sensor networks are “highly flexible...and can be easily adapted to the considered problem. Ex.1006, 2. Tsetsos explains that WSNs adapt to different problems by “**topological** change.” Ex.1006, 2. Ex.1003, ¶224. 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. Ex.1006, 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>
```

Ex.1006, Listing 1, p. 7 (annotated); Ex.1003, ¶224.

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

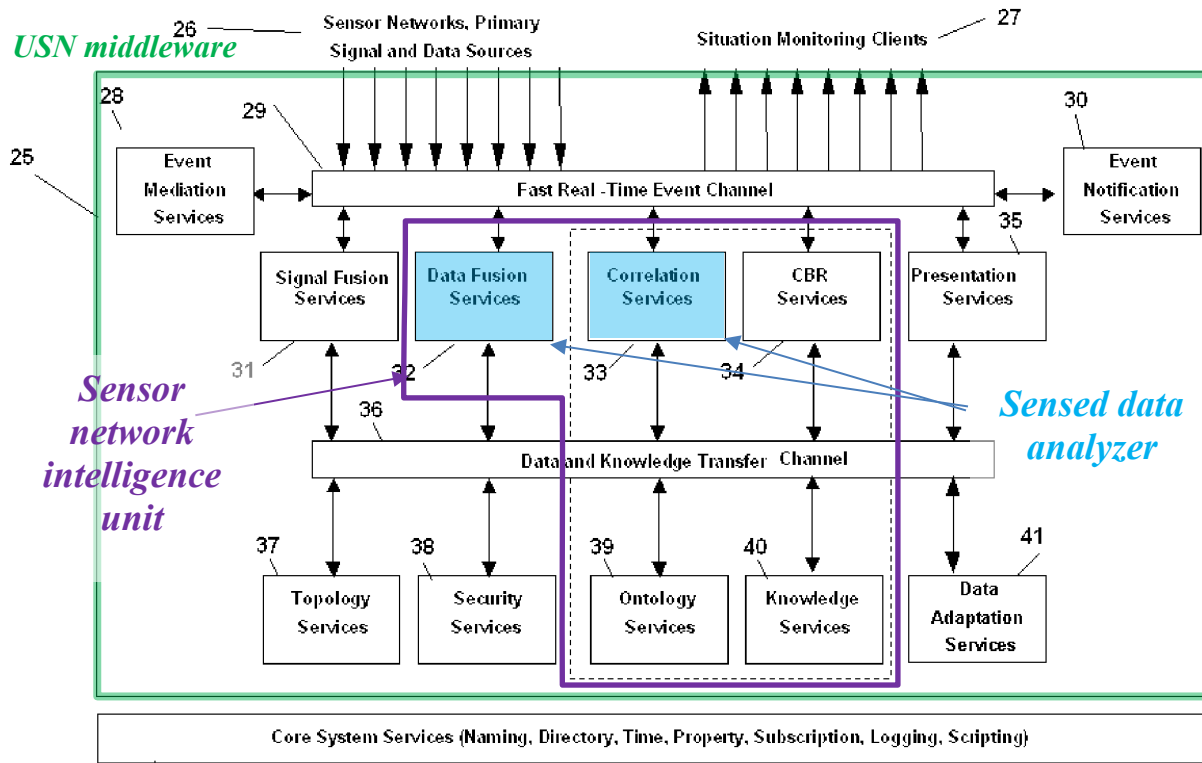
adaptation. Ex.1003, ¶¶225-227.

Thus, Jakobson, alone or in combination with Tsetsos, discloses or renders obvious limitation [3.2]. Ex.1003, ¶228.

6. 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;*

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

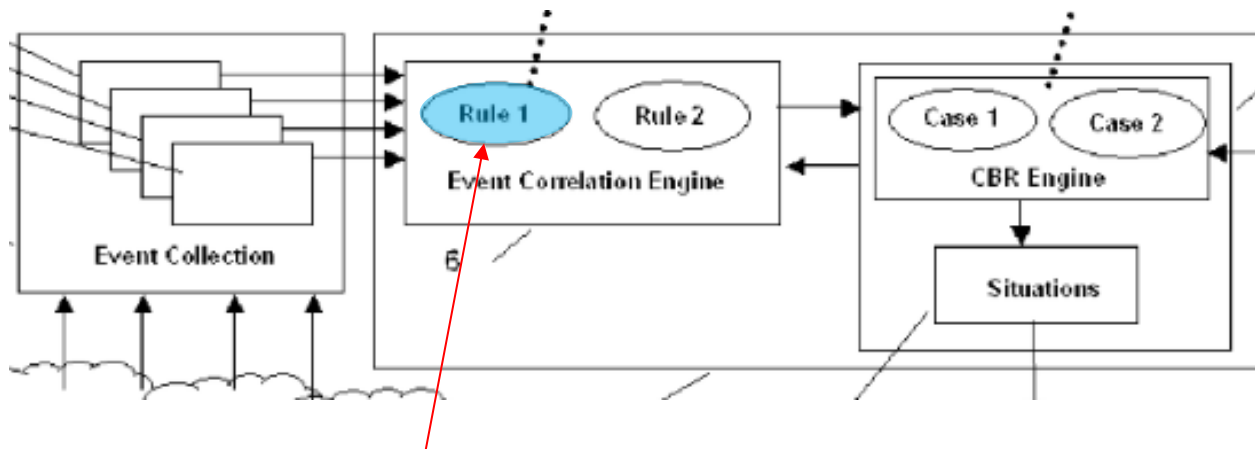


Ex.1005, FIG. 7 (annotated); Ex.1003, ¶229.

Thus, as shown in annotated FIG. 7 above, Jakobson's data fusion module, correlation services module, or both, disclose the *sensed data analyzer*. Ex.1003, ¶229.

Additionally, as illustrated in FIG. 6, the *sensed data analyzer* may include a submodule or engine of the event correlation module. Jakobson explains that the event correlation module uses "one or more event correlation engines" to perform event correlation operations like compression, filtering, suppression, counting, escalation, generalization. Ex.1005, [0013], [0185]-[0187].

One or more of these rules and their associated engine(s) discloses a *sensed data analyzer*, as illustrated below. Ex.1003, ¶230.



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

Ex.1005, FIG. 6 (partial annotated); Ex.1003, ¶230.

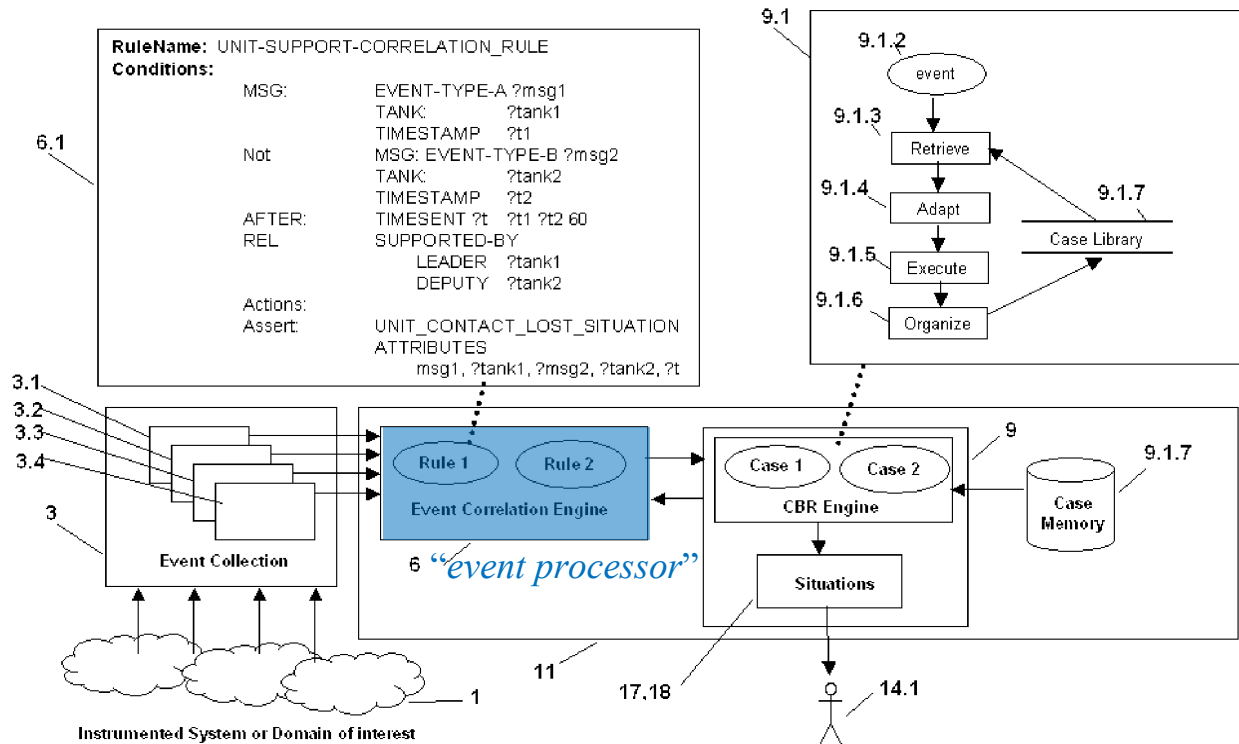
Thus, the data fusion module, the event correlation module, and/or one or more rule modules within the event correlation module, discloses or renders obvious the *sensed data analyzer for performing an analysis operation to clean, classify and integrate the sensed data processed by the sensor network abstraction unit*. Ex.1003, ¶231.

[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

As explained above for limitation [1.4], Jakobson describes multiple examples of “*generating conditional events*,” including the event correlator

generating high-level inferred events.

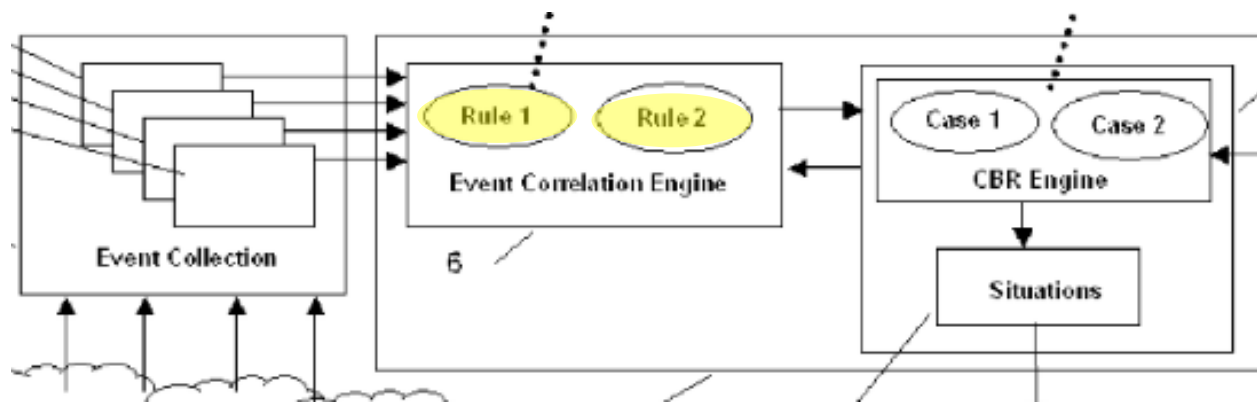
With respect to the high-level correlated events (*the conditional events*), as explained above for limitations [1.4] and [2.1], Jakobson's event correlation module (*event processor*) creates higher-level events based on lower-level events, thresholds, Boolean operators, and conditional predicates, which discloses “*generating the conditional events.*” See Ex.1005, [0019] (inferring higher-level events), [0188] (generating events by thresholding and event clustering based on Boolean operators and conditional terms); [0175] (same), [0176] (thresholding). Ex.1003, ¶232.



Ex.1005, FIG. 6 (annotated); Ex.1003, ¶232.

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 Ex.1005, FIG. 2, [0176], [0191], [0012].

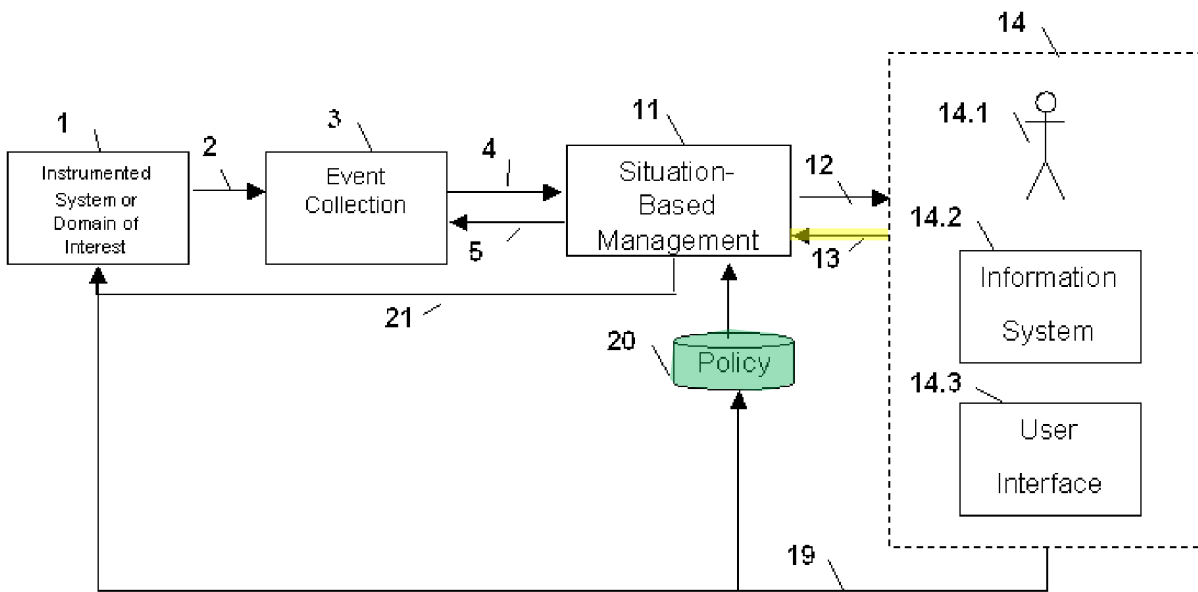
Jakobson explains that these higher-level events, including the notifications of the higher-level status of some condition, are governed by “correlation rules” (*rules*) running on a rule engine at the event correlation module. Ex.1005, [0185].



Ex.1005, FIG. 6 (partial annotated); Ex.1003, ¶234.

Accordingly, the notifications of the properties of the high-level events (*notification and alarm*) are generated *according to a rule*. Ex.1003, ¶¶233-235.

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. Ex.1005, [0266]. Jakobson’s situation viewer communicates with the situation-based management through communications channel 13 (yellow), including issuing “remote procedure calls” and “function calls.” Ex.1005, [0178].



Ex.1005, FIG. 4 (annotated); Ex.1003, ¶236.

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.” Ex.1005, [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. Ex.1003, ¶¶236-237.

It would have been obvious for the situation viewer information service or user interface application (*application program*) to configure or “set” event correlation rules that produce the *conditional events* discussed above. Ex.1003, ¶238. For instance, it would have been obvious for the *application program* to

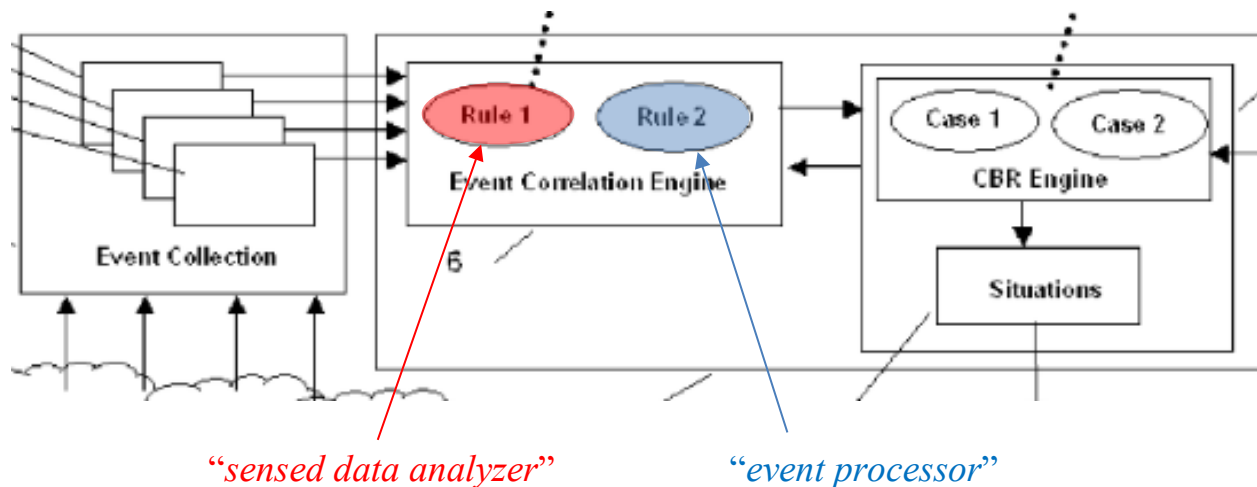
enable the user to control, adjust, or fine tune aspects of event correlation using remote procedure calls or function calls, such as to create or modify (*set*) event correlation rules. Ex.1003, ¶238. For instance, a user viewing the situation reports through the situation viewer user interface may identify issues with the event correlation operations or may wish to customize or optimize event correlation for a given domain or application. Ex.1003, ¶238. Indeed, Jakobson teaches that “there may be various means to control the presentation” and “the situation viewer 14 may also operate on situation-based management 11 via communications channel 13,” including “changing properties of situations, “modifying the ontology of the situation manager.” Ex.1005, [0179]-[0180]. According to Jakobson, “[t]he 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.” Ex.1005, [0180]; Ex.1003, ¶238.

Further, Jakobson explains that the event correlator’s objectives are “to **direct** the event collection and **event distribution mechanism**; and to permit the situation manager to **control or coordinate the EC**.” Ex.1005, [0023]-[0024]. Jakobson explains that the event correlation module is responsible for providing those created events to the situation manager. Ex.1005, [0209]. Thus, the event correlator “*manag[es] the generated events*.” Ex.1003, ¶239.

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*.” As also explained above, the event correlation module may also be mapped to the *event processor*. In the case that the event correlation module is mapped to both the sensed data analyzer and the event processor, it would have been obvious that one set of event correlation rule engines performing the *cleaning, classifying, and integrating* event correlation actions would be the *sensed data analyzer*, and another rule engine (or set of rule engines) generating the *conditional events* would be the *event processor*. Ex.1003, ¶240. This is illustrated below with respect to FIG. 6.

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



Ex.1005, FIG. 6 (partial annotated); Ex.1003, ¶240.

Thus, the event correlation service, or a rule engine running at the event correlation module of Jakobson, discloses or renders obvious 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.*

Ex.1003, ¶241.

[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.*

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*).

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*). Ex.1005, [0168]-[0170], [0181].

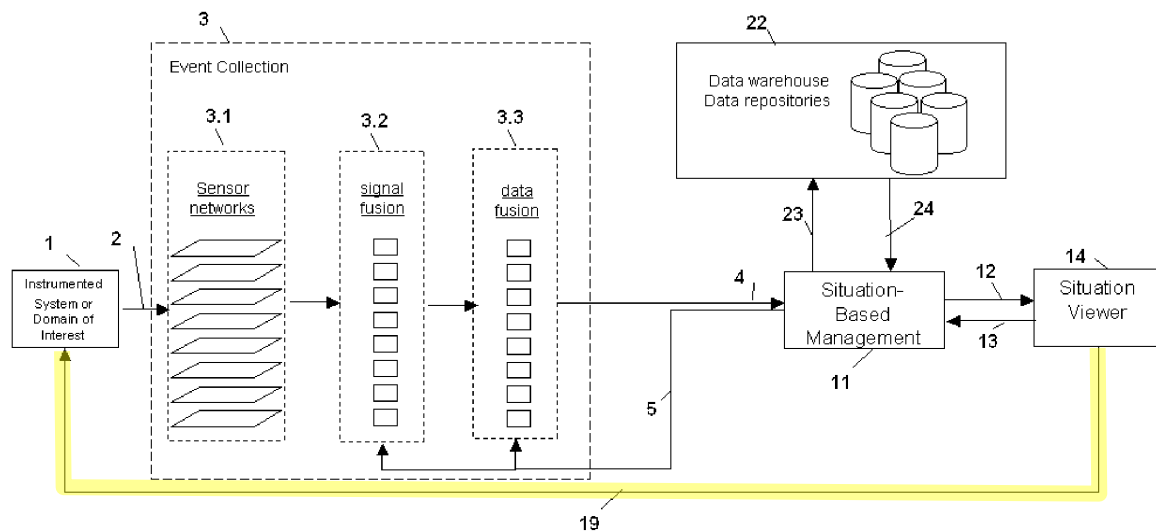
Thus, Jakobson's situation manager (*context information manager*) discloses or renders obvious this limitation. Ex.1003, ¶¶242-245.

7. 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;

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. Ex.1005, [0216]. Jakobson further explains that the situation viewer 14 uses a communication channel 19 (highlighted in yellow below) to “**take action** on the domain of interest.” Ex.1005, [0180]. As explained above at limitation [1.0], the domain is instrumented with the sensors of the *sensor network*. Ex.1003, ¶246.



Ex.1005, FIG. 5 (annotated); Ex.1003, ¶246.

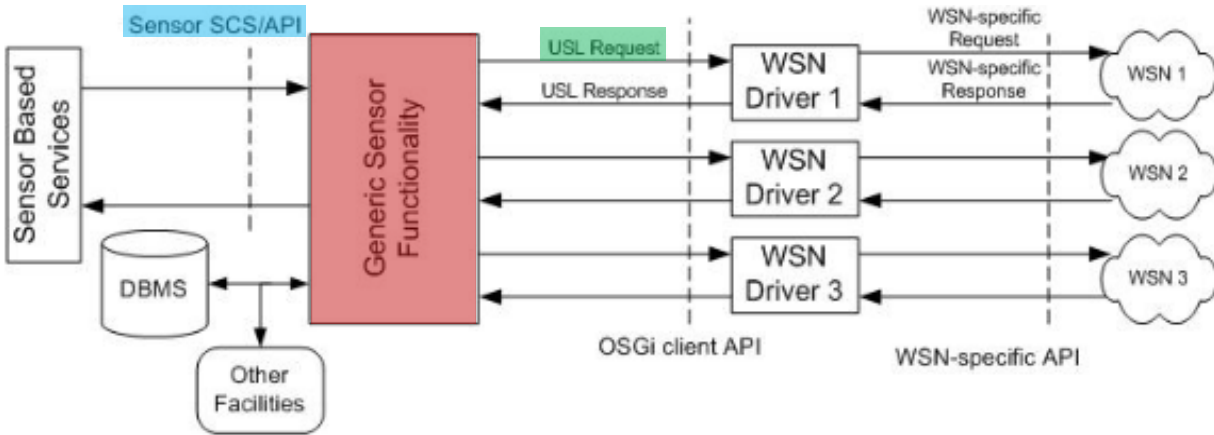
Jakobson also discloses that the situation viewer 14 includes an “information system 14.2” which communicates with the situation-based management 11 “to exchange information such as messages, **remote procedure calls**, and shared

databases.” Ex.1005, [0178].

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. Ex.1003, ¶¶247-248.

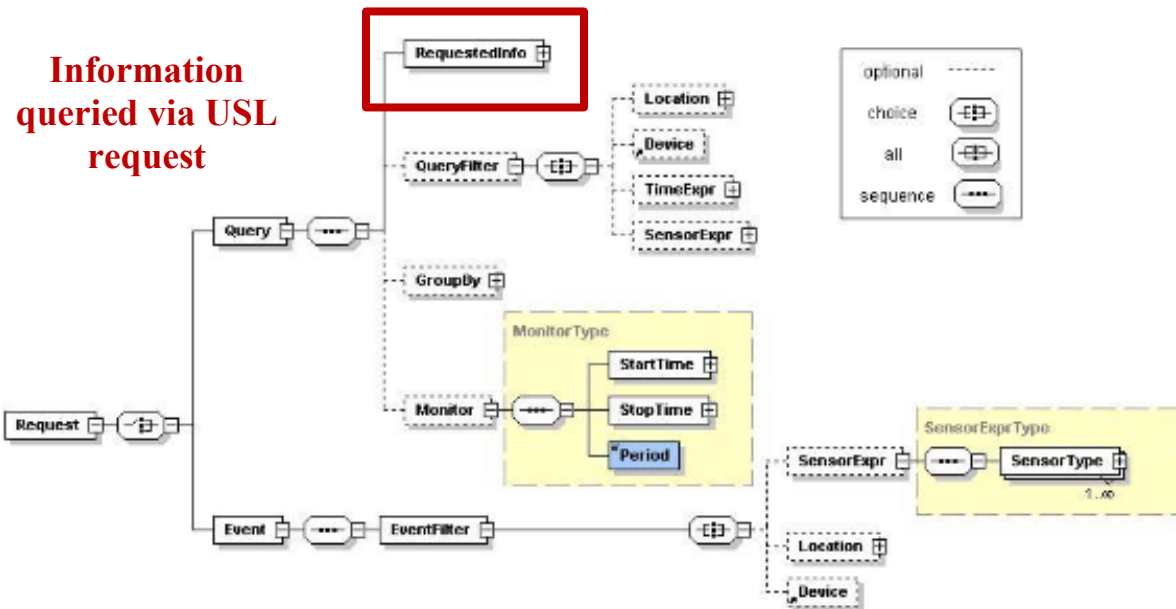
Likewise, Tsetsos also describes functionalities and protocols to issue commands and request information from sensors, such as sensor type, location, and status (*identifying and searching resources of the sensor network*). See Ex.1006, 5-7 (describing USL and GSF commands and requests). For instance, 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). Ex.1006, 4-6; Ex.1003, ¶¶249.

GSF receives SCS commands and issues USL requests



Ex.1006, FIG. 4 (annotated); Ex.1003, ¶249.

Tsetsos explains that these APIs can be used to get information about sensors, including *identifying and searching for* specific sensors and associated information. Ex.1006, 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.” Ex.1006, 6. Further, Tsetsos explains that the GSF API provides “catalogs for discovery of available sensor networks.” Ex.1006, 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. Ex.1003, ¶250.



Ex.1006, FIG. 5 (annotated); Ex.1003, ¶250.

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>
```

Ex.1006, Listing 1, p. 7 (annotated); Ex.1003, ¶251.

It would have been obvious for the situation viewer 14 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*). Ex.1003, ¶¶252-253.

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

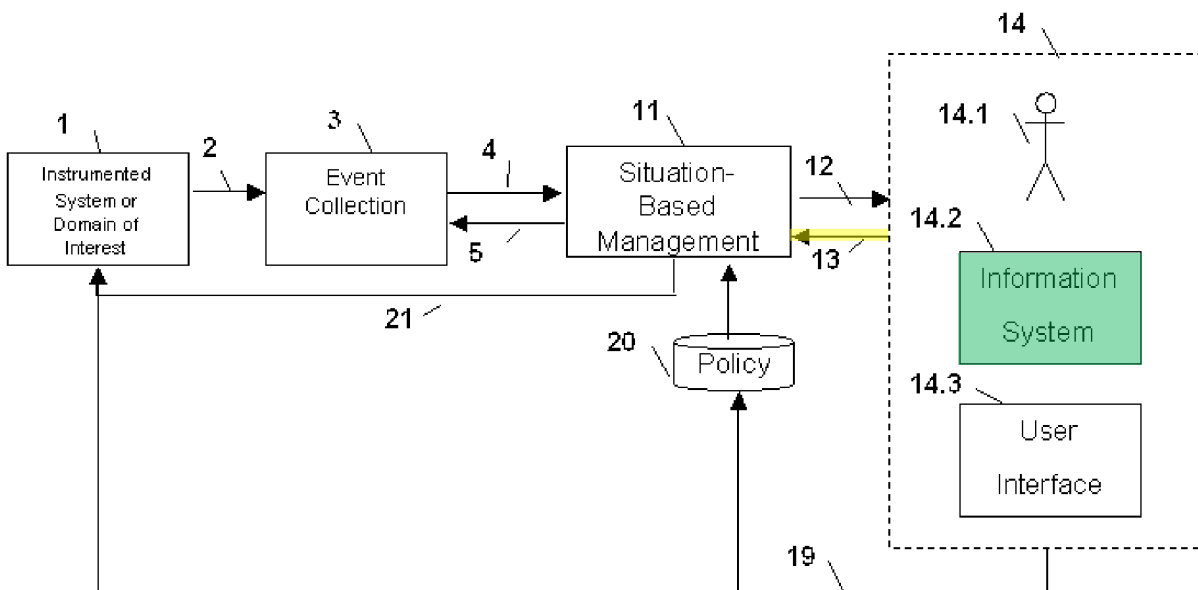
[5.1] a service process manager for scheduling and managing an operation sequence of the sensed data analyzer, the event processor, the context information manager, and the directory service manager in order to obtain a desired information service according to the application program; and

Jakobson explains that its event correlation and situation management techniques include sequences of operations. See Ex.1005, [0079] (SM steps are “sequential and mutually temporally non-overlapping”), [0273]-[0278] (FIG. 15

shows sequential, non-overlapping steps), [0279]-[0280] (“sequences of operations”).

The operational sequences for situation management “might be realized on a multiprocessor computer or using distributed processing in a data processing network.” Ex.1005, [0280]. Accordingly, Jakobson discloses that SBM includes *operation sequence[s]* controlled by computer processing circuitry and software. Ex.1003, ¶¶255-256.

Jakobson’s situation viewer 14 can include an information system 14.2 (green) that communicates with the situation-based management 11 through communications channel 13 (yellow), including issuing “remote procedure calls,” using “polling,” or using “function calls.” Ex.1005, [0178].



Ex.1005, FIG. 4 (annotated); Ex.1003, ¶258.

Jakobson explains that the situation viewer 14 communicates with the situation-based management to control various aspects of situation management, including “changing properties of situations, creating new situations, modifying the knowledge base of situation-based management, and modifying the ontology of the situation manager.” Ex.1005, [0180]. The situation viewer 14 also manipulates policies “that **affect the behavior of the situation manager**, including the **situation evaluation criteria**” in order to obtain a desired information service. Ex.1005, [0180].

Thus, Jakobson discloses that the situation viewer can issue commands and set policies for (*for...managing...*) situation-based management operations (*an operation sequence*). Ex.1003, ¶¶257-260.

It would have been obvious for the SBM middleware 25 to include a module (*service process manager*), similar to the policy repository, for instance, that manages the sequence of operations used by the various inter-dependent modules of SBM to ensure that the operations occur at the correct times (e.g., in a “non-overlapping” fashion). Ex.1005, [0280]; Ex.1003, ¶262. For instance, it would have been obvious for a policy repository to receive remote procedure calls, polling, or other function calls to manage *an operation sequence* of the software modules, rules, and engines of the data fusion module, event correlator, the situation manager, and directory service (*the sensed data analyzer, the event*

processor, the context information manager, and the directory service manager).

Ex.1003, ¶263.

For instance, because the situation viewer 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 policy repository to manage *operation sequence* policies that can be managed, adjusted, and fine-tuned by the user of the situation-based management based on the situation reports the user views. Ex.1003, ¶¶263-264. 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. Ex.1003, ¶¶265-266.

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.*

As explained above at limitation [5.1], Jakobson discloses that the situation viewer 14 can issue a “remote procedure call” to “interface to” SBM to trigger or request situation reports. Ex.1005, [0178]. The remote procedure call functionality discloses *an application program interface*. 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 Ex.1006, 5-7. It would have been obvious to use an API (*application program interface*) to trigger or configure the provision of situation reports (*for providing the application program...*), as Tsetsos does. Ex.1003, ¶¶268-269. 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. Ex.1003, ¶269.

Jakobson discloses that 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*) may be “converted to a different format” (*in an information service form suitable for the application program*) before delivering the report to the information system 14.2. Ex.1005, [0178]; Ex.1003, ¶270.

Further, referring to FIG. 7, Jakobson describes using software modules such as Data Adaptation services 42 and Presentation services 35 (*an application programming interface*) to “support the user interface,” which controls how the situation reports are presented. Ex.1005, [0218]. Ex.1003, ¶¶268-272.

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 14 (*in*

an information service form suitable for the application program). Ex.1003, ¶270.

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.

Ex.1005, [0179].

Thus, Jakobson discloses presenting results processed by multiple components in the SBM hierarchy, including both situations and events. Ex.1003, ¶¶272-273. 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 data fusion and/or corresponding event correlation rule engines (*sensed data analyzer*), high-level inferred events from the

event correlator (*event processor*), situations and 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*). Ex.1003, ¶¶273-276. 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. Ex.1003, ¶¶275.

Thus, Jakobson discloses this limitation, or renders it obvious alone or in view of Tsetsos. Ex.1003, ¶277.

8. Claim 6

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. Ex.1003, ¶278.

[6.0]⁷

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 heterogenous sensor networks*).

⁷ For claims 6-12, the claim language for each numbered limitation can be found in the Listing of Claims beginning on p. 6 of this Petition.

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. Ex.1003, ¶¶279-280.

[6.1]-[6.3]

Jakobson renders obvious limitations [6.1], [6.2], and [6.3] for the same reasons explained above for limitations [2.0], [2.1], and [2.2], respectively. Ex.1003, ¶¶281-283.

9. Claim 7

Jakobson and Tsetsos render obvious limitations [7.0], [7.1], and [7.2] of claim 7 for the same reasons explained above for limitations [3.0], [3.1], and [3.2], respectively. Ex.1003, ¶¶284-286.

10. Claim 8

Jakobson and Tsetsos render obvious limitations [8.0], [8.1], and [8.2] of claim 8 for the same reasons explained above for limitations [4.0], [4.1], and [4.2], respectively. Ex.1003, ¶¶287-289.

11. Claim 9

Jakobson and Tsetsos render obvious limitations [9.0], [9.1], and [9.2] of claim 9 for the same reasons explained above for limitations [5.0], [5.1], and [5.2], respectively. Ex.1003, ¶¶290-292.

12. Claim 10

[10.0]

Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.3] (*USN middleware*) 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*. Ex.1003, ¶293.

[10.1]

Jakobson and Tsetsos render obvious this limitation for the same reasons explained above for limitations [2.0], [3.0], and [3.2]. Ex.1003, ¶294.

[10.2]

Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [1.3], [2.1] and [4.0]. Ex.1003, ¶295.

[10.3]

Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitation [2.1]. Ex.1003, ¶296.

[10.4]

Jakobson discloses or renders obvious this limitation for the same reasons explained above for limitations [2.1] and [4.1] (*generating an event including*

notification event and alarm...according to a rule set by an application program), and [1.4] (*knowledge contents*). Ex.1003, ¶297.

[10.5]

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*). Ex.1003, ¶298.

[10.6]

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*). Ex.1003, ¶299.

13. Claim 11

[11.0]

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

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. Ex.1003, ¶301. 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. Ex.1005, [0054], [0184], FIG. 7; Ex.1003, ¶301. Tsetsos similarly shows that the middleware receives data from multiple wireless sensor networks (WSNs). Ex.1006, 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* Ex.1005, [0215]; Ex.1006, 6 (describing unified handling of sensor requests and sensor data). Ex.1003, ¶301

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

14. Claim 12

[12.0]

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

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*.”

Because these timing elements are recited in the alternative, the prior art need only disclose or render obvious one of those options. Ex.1003, ¶¶304-305. However, to the extent Jakobson does 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. Ex.1003, ¶305.

For instance, Jakobson discloses updating situations “*periodically*.” Ex.1005, Abstract. Further, Jakobson discloses that the information system can issue a “remote procedure call” and be provided a “situation report.” It would have been obvious to a POSITA for the remote procedure call to be used to trigger a single situation report (*only one time*). Ex.1005, [0178]. Ex.1003, ¶¶306-307.

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*). Ex.1003, ¶307.

Accordingly, Jakobson discloses or renders obvious this limitation alone, or

in combination with Tsetsos.

X. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. No basis for *Fintiv* denial

The *Fintiv* factors favor institution. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential).

Factor 1: No motion to stay has been filed, and thus this factor is neutral.

Factor 2: The Board will likely issue a Final Written Decision in July/August of 2026, and the district court trial date is currently set for March 16, 2026. Ex.1011. The estimated trial date based on time-to-trial statistics is May 2026. Ex.1012, 5 (21.9 months). This factor is neutral because the trial date is speculative and subject to change. *See* Ex.1013, 8 (Interim Procedure For Discretionary Denials) (“scheduled trial dates are unreliable and often change”). For example, Judge Gilstrap has *eleven* trials set for March 16, 2026. Ex.1014. Further, a Motion to Dismiss for Lack of Venue is currently pending before the Court.

Factor 3: The co-pending litigation is in its early stages, and the investment in it has been minimal. The claim construction hearing is not until September 16, 2025, fact discovery is not set to close until October 17, 2025, and expert discovery is not set to close until December 4, 2025. Ex.1011, 1-2. Moreover, Petitioner only learned which claims were being asserted on November 14, 2024, and has worked

expeditiously to file this petition within three months. Ex.1015 (infringement contentions cover pleading). This factor favors institution.

Factor 4: Final invalidity theories will not be submitted until March 11, 2026, so the extent of overlap is currently speculative. If Patent Owner raises a *Fintiv* challenge in its preliminary response, Petitioner will evaluate whether a stipulation is appropriate based on the state of the district court proceeding at that time. This factor is neutral.

Factor 5: Petitioner is a defendant in the co-pending litigation, and thus this factor is neutral.

Factor 6: This Petition presents compelling evidence of unpatentability, and thus denial is inappropriate. Ex.1013, 3-5. The Petition—supported by Dr. Houh’s expert testimony—plainly shows that the ’232 patent claims no more than what was already known or obvious.

In sum, every *Fintiv* factor is either neutral or weighs against denial.

B. No basis for §325(d) denial

Denial under 325(d) is not warranted because the challenges presented in this petition are neither cumulative nor redundant to the prosecution of the ’232 patent. The Examiner did not consider any of the references applied in the above grounds.

XI. CONCLUSION

Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Dated: February 21, 2025
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, Texas 75201
Customer No. 27683

Respectfully submitted,
/Scott T. Jarratt/
Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297

XII. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. §42.8(b)(1), Petitioner certifies that the real parties-in-interest are Amazon.com, Inc., Amazon.com Services LLC, and Amazon Web Services, Inc.

B. Related Matters

Pursuant to 37 C.F.R. §42.8(b)(2), to the best knowledge of the Petitioner, the '232 patent is or was involved in the following case:

Case Heading	Number	Venue	Filed
<i>KAIFI LLC v. Amazon.com, Inc. et al</i>	2:24-cv-00542	EDTX	Jul. 17, 2024

C. Lead and Back-up Counsel and Service Information

Lead Counsel

Scott T. Jarratt
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (972) 739-8663
Fax: (214) 200-0853
scott.jarratt.ipr@haynesboone.com
USPTO Reg. No. 70,297

Back-up Counsel

Brian Kwok
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (415) 293-8916
Fax: (214) 200-0853
brian.kwok.ipr@haynesboone.com
USPTO Reg. No. 58,828

Brett A. Bostrom
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (720) 484-3715
Fax: (214) 200-0853
brett.bostrom@haynesboone.com
USPTO Reg. No. 73,009

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,943. Pursuant to 37 C.F.R. §42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: February 21, 2025

/Scott T. Jarratt/
Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. §42.6(e) and 37 C.F.R. §42.105, service was made on Patent Owner as detailed below.

Date of service February 21, 2025

Manner of service FEDERAL EXPRESS

Documents served Petition for *Inter Partes* Review Under 35 U.S.C. §312 and 37 C.F.R. §42.104 of U.S. 8,040,232; Petitioner's Power of Attorney; Petitioner's Exhibit List; Exhibits Ex.1001-Ex.1020.

Persons served Broadview IP Law, PC
3200 El Camino Real, Suite 260
Irvine, CA 92602

/Scott T. Jarratt/

Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297