Towards context-aware supervision for logistics asset management
Concept design and system implementation

Feng, Fan; Pang, Yusong; Lodewijks, Gabri

DOI
10.1007/978-3-319-53076-5_1

Publication date
2017

Document Version
Final published version

Published in
Information Technology for Management: New Ideas and Real Solutions

Citation (APA)
Feng, F., Pang, Y., & Lodewijks, G. (2017). Towards context-aware supervision for logistics asset management: Concept design and system implementation. In E. Ziemba (Ed.), *Information Technology for Management: New Ideas and Real Solutions: Proceedings of the 14th Conference, AITM 2016 and 11th Conference, ISM 2016, Revised Selected Papers* (pp. 3-19). (Lecture Notes in Business Information Processing; Vol. 277). Springer. https://doi.org/10.1007/978-3-319-53076-5_1

Towards Context-Aware Supervision for Logistics Asset Management: Concept Design and System Implementation

Fan Feng(✉), Yusong Pang, and Gabriel Lodewijks

Transport Engineering and Logistics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
{f.feng,y.pang,g.lodewijks}@tudelft.nl

Abstract. Innovations of information and communication technology (ICT) open plenty opportunities to promote internal operation efficiency and external service level in logistics. As current logistics developments tend to be more complex in operation and large in scale, recent practices start to pay more attentions on improving asset (e.g. equipment and infrastructure) management performance with new ICT development. One of the primary concern is to improve system robustness and reliability. It not only requires the supervision system be capable of diagnosing the condition of the system, but also proficient to find the intrinsic relationship between different conditions and resources thus lead to an integrated decision making process. Moreover, recent ICT innovations, such as WSN and IOT, could record and deliver system descriptors (physical measurements, virtual resources, operational configurations) in real time. Such large-stream and heterogeneous data requires an integrated framework to process and management. To address such challenges, in this paper, a novel concept of context-aware supervision is proposed. An intelligent system with integration of semantic web and agent technology is developed, which aims at providing condition-monitoring and maintenance decisions to relevant user. A generic ontology-agent based framework will be illustrated. The developed system will be applied for the supervision of a large-scale material handling system-belt conveying system as a proof-of-concept.

Keywords: Logistics asset management · Context-awareness · Ontology-agent integration · System supervision

1 Introduction

As the scale of transport and logistics system tremendously expanded, the attention of researchers and engineers have been shifted from improving operational efficiency towards enhancing system reliability. Takata [4] revealed this paradigm shift by demonstrating that the ultimate objective for a manufacturing is not only to produce products in an efficient way but also to provide the functions needed by society in a sustainable and reliable way. It is well known that operations of large logistics infrastructures are heavily rely on reliable and efficient equipment, such as container transportation, production assembling line and etc. And such equipment, like automated

E. Ziemba (Ed.): AITM 2016/ISM 2016, LNBIP 277, pp. 3–19, 2017.
DOI: 10.1007/978-3-319-53076-5_1

assembling line machinery, AGV and belt conveying system, are normally have complex structures and massive capital investment. If a malfunction of single component or process has not been detected and corrected timely, it could lead to an expensive downtime and furthermore impose a great impact on the entire logistic activities. Consequently, a system with decision support becomes an essential element to provide relevant users a consistent understanding regards the system status and enabling an effective planning and execution of maintenance.

To implement an efficient supervision system, ICT is seen as a key enabler. Indeed, the ICT has been widely used for assisting information intensive tasks in logistics such as resource planning, order processing, crew scheduling and so on [1–3]. With respect to asset management, it demands an efficient supervision system to delivering right information (e.g. system condition, maintenance decisions) to right person (inspector, technical engineer) at right time to right place. Specifically, how to determine the 'rightness' with respect to fault condition identification, information integration, information presentation and information delivering becomes critical. Several challenges are summarized as follows:

- Heterogeneity: The heterogeneity is defined as system entities have different types of data model, properties, operation mechanisms and even different hardware and operating system [5]. As for TEL management especially for asset management and supervision, different data resources, operational information, past experiences and knowledges are characterized as heterogeneous resource, thus impose difficulties in integration.
- Interoperability: With respect to interoperability, three perspectives can be identified [6], (1) organizational level: generic approaches and shared understanding of concepts, process, beliefs and terms [7]. (2) system level: interconnection between independent systems. (3) data level: consider the data properties include data format, data availability, data representation and semantic meanings. With respect to asset supervision, the interoperability challenges are inevitably presented at all three levels.
- Integrated decision making: Logistics asset normally depicted as large-scale and complex equipment. If a malfunction of single component or process has not been detected and corrected timely, it could lead to an expensive downtime and furthermore impose a great impact on the entire logistic activities. Consequently, a system with decision support becomes an essential element to provide relevant users a consistent understanding regards the system status and enabling an effective planning and execution of maintenance, such functionality could be referred as integrated decision making.

To confront with above mentioned challenges, the concept of e-maintenance which utilizes ICT technology to implement a web-based monitor system is suggested [8]. Candell et al. [9] presented an e-maintenance framework that integrated maintenance and ICT perspectives to support maintenance task in aviation industry. Arnaiz et al. [10] applied e-maintenance in manufacturing and logistics operations by means of advanced ICT technologies. Though attractive, the e-maintenance system still has limitations [11]: (1) instead of introducing a layer of semantic, the use of e-maintenance system focused on expanding the descriptive scope of existing models which results in an inefficient support for format generalization and standardization and eventually too complex for users to process; (2) direct working on large stream of data could limit contributions to share knowledge and expertise. Followed by e-maintenance, a new paradigm has emerged named context-aware monitoring system [12]. It was firstly designed as a component of ubiquitous computing environment and recently been introduced to the system monitoring domain. Galar et al. [13] presented a hybrid context-awareness model to facilitate the asset life cycle management. Evchina et al. [14] proposed a context-aware middleware to assist the data-intensive monitoring tasks, and the framework is validated by means of case study in the domain of building automation. Pistofidis et al. [15] proposed a context-awareness system for asset management based on mobile and cloud technology. Inspired by the concept of context-aware monitoring, in this paper, the concept of context-aware supervision will be initiated.

The works presented in this paper is a continues work with respect to [36]. In [36], the concept of CASS is briefly introduced. In this paper, we elaborate the detailed implementation of the CASS. The purpose of CASS will be further motivated. The developed system will be applied to assist the practical maintenance tasks for a large-scale belt conveyor system. Moreover, the software packages used to system development will be specifically explained.

The remaining part of the paper is organized as follows: Sect. 2 will introduce the concept of context-aware supervision system (CASS) and the motivations behind it. Section 3 will provide key technological enablers that could support the implementation of a CASS system. Section 4 will first present the system design with an abstract structure perspective and the design of each functional block. Section 5 will present a case study of applying CASS for intelligent supervision of a large-scale belt conveying system. The conclusion and future works will be addressed in Sect. 6.

2 The Concept of Context-Aware Supervision

The definition of context is given by [16] as *any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.* And a system can be termed as context-aware if *it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task* [16]. A context-aware system (CAS) adapts and provides relevant information and the most appropriate service to users in an active and autonomous manner while requires little interactions [13]. Take advantage of such property, the CAS has been applied in various domains such as health care [17], pervasive computing [18], data intensive monitoring [14], enterprise application model [19] and so on. It is noted that the mentioned CAS are represented with a layered framework which is generally composed of sensor layer for data acquisition, data layer for storage, processing layer for context modelling and application layer for context delivery and representation. The ultimate goal is to deliver right information with right format to right user in right time.

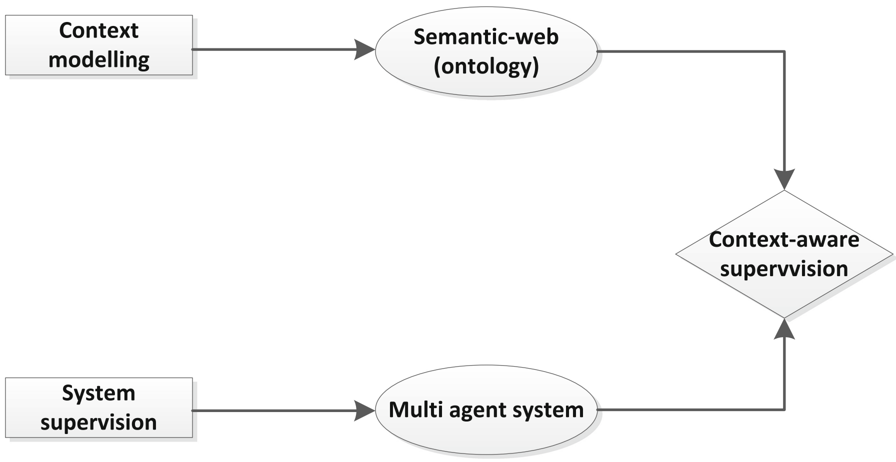

Fig. 1. General philosophy of a context-aware supervision framework

In this research, we focus on investigating the potential of applying CAS for asset supervision service. In essence, the supervision services include system monitoring, failure/abnormality diagnosis/prognosis and maintenance planning. Apart of being context-aware of delivering meaningful information to user, it also requires a transparent flow from data to supervision method. The objective of a supervision system is to deliver accurate and timely information regards system conditions and propose effective maintenance actions to ensure reliability and availability of the system. Its success relies on integrating different diagnosis/prognosis methods and such methods most often have certain scope of applicability and input context. As such, it put additional requirements on CAS to systematically integrate and manage system supervision processes. To integrate CAS for system supervision and fulfil additional requirements, a novel concept of *Context-aware supervision* is initiated as:

A context-aware supervision system (CASS) should include a series of functionalities include monitoring, supporting and advising in relation to system events. It not only focuses on diagnosing and prognosis failures, but also responsible for managing and organizing system knowledge, reasoning facts, integrating resources and analysing problems. As such, the failure context can be given in a more meaningful manner that delivers information include: the specification of fault condition, recorded data linked to it, maintenance or operating actions linked to it, users that responsible for it and method that been used to determine it.

The characters of *context-awareness* are presented at two levels in CASS: (1) the supervision method should be aware of the context information it operates upon; (2) the end user should comprehend the created supervision context. In literature, several works have been established which attempt to consolidate the concept of context-aware with asset management such as context-awareness predictive maintenance [20], context-aware E-maintenance [15] and context-aware condition monitoring [13]. Two limitations are drawn from previous works: (1) the scope of applicability of proposed concept is limited given the fact that it only concerns partial aspect of the supervision process, For instance, the work of [20] concerns predictive maintenance, it lacks details regards how context been modelled and processed, (2) most works stay on a conceptual level which lack of sufficient technical details on how to put the concept into practice.

A context-aware supervision system is used for information integration and supervision of large-scale asset service in TEL domain. The key ICT enablers are semantic web and autonomous agent and its philosophy is depicted in Fig. 1. Specifically, the ontology is used to model the semantic connections for heterogeneous data and various entities in supervision domain, thus enable information integration, data filtering and problem decomposition for specific supervision tasks. The use of agent system intends to provide intelligent diagnosis and decision making functionalities through agent intelligence and cooperation. Our work contribute to the literature by first introducing the concept of CASS, then we will discuss the technology been selected for putting such concept into action. Finally, a case study would demonstrate how the system works.

3 Key Methodologies for CASS Implementation

This section will discuss the key technological enablers for implementing a CASS, which include the method for context-modelling and system supervision.

3.1 Context-Modelling Methodology

Intuitively, large amount of context information is either acquired or derived from sensor devices. Normally, there exist gaps between raw data and the level of information which is useful to applications [21]. The context-modelling is used to bridge this gap by processing and transforming raw data before passed to context-aware services. Krummenacher et al. [22] have proposed several criteria for context model selection which include applicability, comparability, traceability, quality and so on. Meanwhile, Hoareau et al. [21] conducted an extensive review for existing modelling choices such as key value models, makeup scheme, logic based models, object oriented models, ontology and so on. According to their in-depth discussion, the use of ontology is proposed to be the most expressive models to fulfil our requirements [23]. A formal definition of ontology is given by [24] as *an explicit specification of a conceptualization* which was used to describe a specific domain knowledge where concepts and relationships are unambiguously defined and checked.

Recent works extensively applied ontology to facilitate the context modelling, its applicability covers domain include risk management of cold chain logistics [25], enterprise application [26], process supervision [27] and so on.

3.2 System Supervision

A system supervision process considers providing users with decision support before/during/after the occurrence of system failure or abnormal situations. A typical supervision process consists of data acquisition, condition diagnosing/prognosis and maintenance planning. In this paper, rather than considering specific method or algorithm, we focus on how to provide a generic and adaptive environment to incorporate and integrate different methodologies and mechanisms operate together with flexibility and scalability. Several requirements are given as below:

- Flexibility: Be aware of and accommodate to different system conditions and retrofits seamlessly.
- Cooperative: It should cooperate and coexist with different software system and third party interface. Be capable of accessing information and providing diagnosis result in meaningful form.
- Collaborative: Enable collaborative decision making process with different diagnosis/prognosis methods. Support seamlessly information exchange between different supervision modules to achieve cohesive judgment of overall system conditions.
- Extensibility: The system should be able to incorporate new supervision method or update existing one without significant change of the system architecture. New measurement, supervision intelligence and human expertise could be deployed seamlessly as needed.

Agent, as a tool in artificial intelligence domain, provides a way of dealing with complex engineering problem and establishing adaptive system for decision making and information management through agent intelligence and collaborations [28]. State-of-arts demonstrate that agent system have been largely applied to support system supervision functions, which include condition monitoring [29], risk management [30] e-maintenance [31] and so on. Given such facts, agent technology is chosen as the key enabler for supervision system design, reasons are given as: (1) agent could cooperate and deploy on top of existing software; (2) in a multi-agent-system, agents could collaborate with each other to communicate and exchange information; (3) agent system could be deployed in distributed environment where new agent could easily join the system or leave the system as needed.

3.3 Ontology-Agent Integration

Key technological enablers have been chosen in previous section. We select ontology as the context-modelling method and agent system as the environment to support system supervision integration. In order to implement CASS, a next step is to consider the integration issues. In literature, several works have been established that concerns the integration of ontology and agent system. Natarajan et al. [32] developed an ontology-agent framework for condition supervision of large chemical plant. Dibley et al. [33] presented a work of building monitoring system where three ontologies are developed to capture the major semantics of a building environment and agent system is deployed to facilitate the monitoring tasks. For most of existing works, attentions are paid on using ontology to assist agent communication and knowledge retrieving. To implement CASS, potentials include information analysis, problem decomposition, agent status control are needed. To achieve this, a novel ontology-agent integrated framework is proposed, it will be elaborated in next section.

4 System Framework

The key framework will be given in this section. We first present a comparison between a context-aware system and the proposed context-aware supervision system. It used to emphasis the purposes and objective of the concept. Then aspects such as context model design, agent system design and ontology-agent integration mechanism will be elaborated in different sections.

4.1 Comparison Between CAS and CASS

Figure 2 presents a comparison between a context-aware system and the proposed context-aware supervision system from an abstract structure perspective. A classical context-aware system follows five key processes [17]: (1) context information acquisition: gather information from virtual resources and physical sensors; (2) context-information persistent: data filtering and storing; (3) context-aggregation/reasoning: interpretation and transfer low-order data to high-level applicable information via aggregation and reasoning; (4) context information utilization/delivering: apply context information to implement application-specific service; (5) context representation.

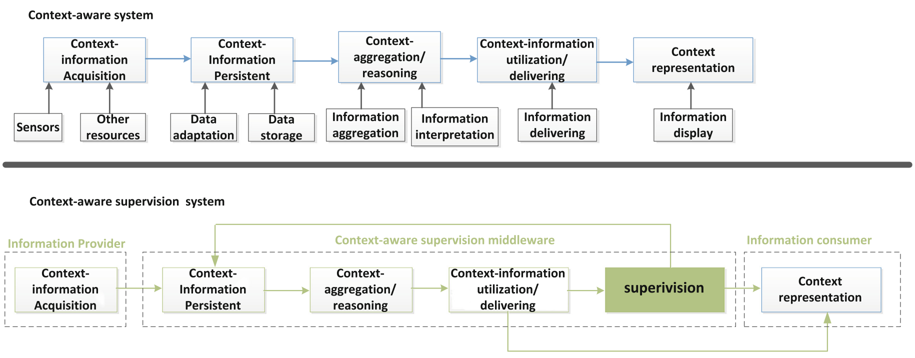

Fig. 2. System abstract architecture: a comparison between CAS and CASS

The abstract structure of a CASS can be found in Fig. 2 as well. All of the functional blocks from CAS are inherited and implemented in CASS. The major difference is distinguished from two aspects:

- The supervision block: The key objective of CASS is to assist system supervision with aspects include equipment monitoring, condition diagnosis and maintenance planning. Such tasks are unable to perform well by only using context-modelling (e.g. ontology reasoning). In most cases, it requires advanced platform/engine for decision making. As such, the supervision block is introduced.

The information flow: The information flow is also adjusted. In CAS, the information flow follows an open loop style where data is gathered from ground layer, processed through each functional block and becomes context-aware. For CASS, a partial closed-loop is formed. In essence, the aggregated and processed context information will be the input source for supervision module. The output of supervision module will be feedback for further processing. It will be first stored in data base and then processed by context model. By doing so, not only the measured data from ground layer would be context-aware but also the supervised result will be aggregated with other relevant information together to make result meaningful to end user. Moreover, it would be helpful to use the returned supervision information to infer new knowledge and propose further actions.

The major abstract module for a CASS is explained, the detailed system design will be given in the following sections.

4.2 Context Model Design

We design an ontology termed *ontoSupervision* to capture major concepts and relationships related to system supervision process. The schematic of *ontoSupervision* is given in Fig. 3 and the explanations of each taxonomy are given below:

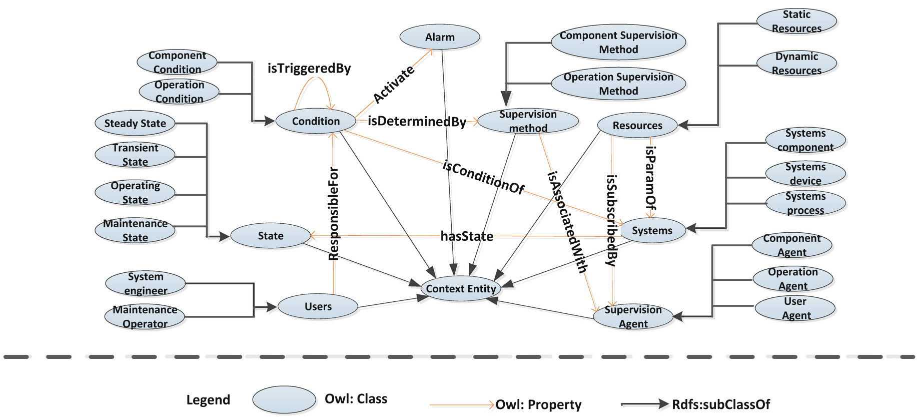

Fig. 3. Upper ontology taxonomy: definition of classes and object properties

System taxonomy is the core concept that presents a description of the system. It includes notions of system fundamental components and operations that needs to be supervised. In addition, the subclass *system devices* contains peripheral devices. Other taxonomies either direct or indirect connect with system through well-defined relationships.

Condition taxonomy incorporates the notion of condition in the system. Two subclasses are included, namely operation condition and component condition. The former one concerns the system abnormal condition during operations and the latter one addresses the physical condition of system at different levels (component, instrument and equipment).

Resources taxonomy represents all relevant information resources that need to be accessed by supervision method. It is composed of two subclasses, namely static resources and dynamic resources. The former one can be thought as the resource that does not change over time such as system specification, historical information and system configurations. The latter one represents the notion of resources that change in real time, such as data acquired from sensor devices and any updated supervision results.

Supervision agent & method represents all available agents been deployed in current system and its associated supervision method. In essence, it serves as a bridge that connects context model with agent system.

State taxonomy represents possible state of the system. In current model, four states are considered namely maintenance state, transient state and operation state and steady state.

Alarm taxonomy represents the notion of possible alarm level that been activated by supervised conditions in the system.

User represents the information consumers in the system. It specifies the responsibilities and point of interests of respective user.

Apart of taxonomy definitions, a complete ontology also includes the design of properties. The object property is used to specify the relationship between two individuals (instance of classes), where the data property is used to link individual to data values. For *ontoSupervision*, the object properties are depicted in Table 1.

Table 1. Object properties of *ontoSupervision*

Property	Domain	Range
isAssociatedWith	Supervision method	Supervision method
isParameterOf	Resources	System
isSubscirbedBy	Resources	Supervision agent
isDeterminedBy	Condition	Supervision method
Activates	Condition	Alarm
hasState	System	State
hasCondition	System	Condition
isTriggeredBy	Condition	Condition

4.3 Multi-agent System Design

Regardless of the models, scopes and design tools, all supervision methods require system measurements as input and transform system conditions (operation and component condition) to supervision result. The supervision result refers to the one that is understandable by respective users. Such common structure allows the supervision methods to be represented as supervision agent [27]. The multi agent system can be perceived as a wrapper which provides environment for different supervision intelligence to perform supervision tasks. It also enables integrated decision making by taking the advantages of agent communication and collaboration. In the proposed framework, three kinds of agent are employed:

- Supervision Agent: Two categories of agent groups are considered as supervision agent. The first one is termed healthiness agent (HA) which is responsible for fault diagnosis at different level of system granularity. Single HA could be used to assess the condition of piece of equipment while multi HAs could work together for evaluating the overall healthiness of the whole system by consolidating different conditions. Another one is termed operation agent (OA) which is used to capture the abnormality during system operation. Typically, it is used to identify the abnormal deviation from normal operations or improper configurations. The agent intelligence, scope of interest, input information and responsibility are determined by its associated methods.
- Information Mediator Agent: The information mediator is used to manage and control the agent execution and interactions. Its necessities are given as (1) It serves as an information portal for supervision agents.; (2) it keeps an active connection between agent system and ontology knowledge model.
- User Agent: It contains the information consumer of the system. Any on-going supervision conditions will be relayed to it via IMA. Sophisticated GUI will connect with it to provide end user a friendly interface.

The overall agent system framework is given in Fig. 4 which follows a *subscription* interaction protocol. The initiator (supervision agent and user agent) sends subscription request to the participator (information mediator agent) indicating its desired information. If the subscribe action success, a permanent communication channel is established between the initiator and participator. The advantage of adapting subscription protocol is given that only information needed would be delivered to target agent with efficiency and accuracy.

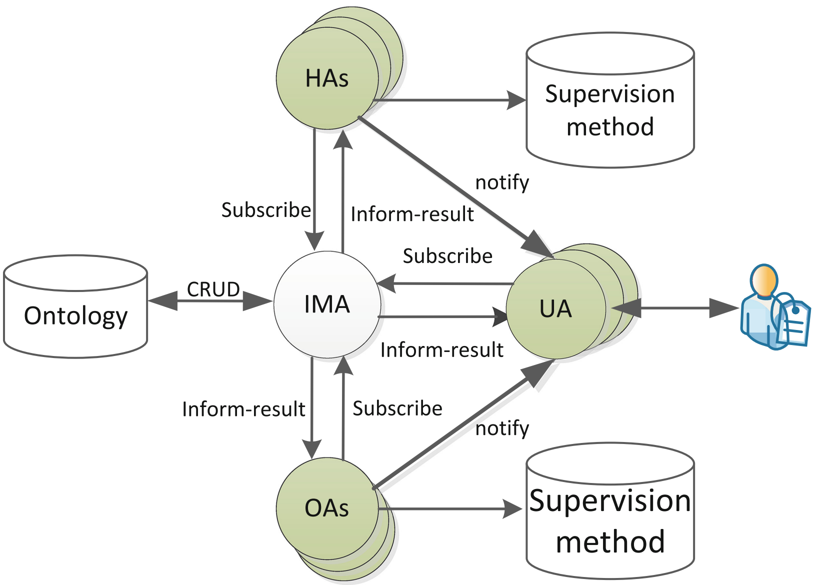

Fig. 4. Agent system framework: topology and communication protocol

4.4 Agent-Ontology Integration

The key of agent-ontology integration is achieved via the interaction between information mediator agent and ontology knowledge base. Such interactions aim at manipulating ontology to acquire information and knowledges where actions include create, read, update and delete entities in ontology. We identified three major processes:

- *Information acquisition*: In this case, ontology is treated as a hybrid database which is used to locate and retrieve information. A typical scenario can be that when a new sensor measurement is available in ontology, the IMA could retrieve it by executing well defined query template. An example of a query template is shown in Fig. 5.

```
SELECT ?par ?agent ?system ?method ?timeStamp
?state ?measurement
WHERE {
            ?par onto:isSubscribedBy ?agent.
            ?par onto:isParametersOf ?system.
            ?agent rdf:type onto:AgentName.
            ?method onto:isAssociatedWith ?agent.
            ?system onto:hasState ?state.
            ?par onto:hasTimeStamp ?timeStamp.
            ?par onto:isMeasuredBy ?measurement.}
```

Fig. 5. Example of information acquisition template

- *Knowledge acquisition*: It fully utilizes the reasoning capability of an ontology model. When certain information is available, the ontology could infer new knowledge by executing context-rules. For instance, if a belt idler temperature is over 70 degree, the ontology could use rules to determine that such context indicates a idler is in fault condition.
- *Knowledge reasoning and agent control*: As discussed previously, a partial closed loop is formed in the structure of CASS. The key motivation is given that any returned supervised information could be further processed by ontology. And the inferred knowledge could be useful in coordinating agent activities. For instance, a misalignment condition often occurs during the running of a belt conveyor and such condition could be induced by multiple reasons (improper power supply, overloading and so on) and such casualty relationships could be pre-defined in the ontology via proper object properties. By doing so, when a misalignment condition is supervised and returned, the ontology could running context rules to find the relevant condition relate with it. Consequently, the associated agent will be activated to allow a depth investigation of the root cause.

5 Case Study

The concept of CASS and its implementation framework has been proposed. Key technology enablers and their integration have been elaborated. To demonstrate how it works, a prove-of-concept is presented in this section. We applied CASS for a large scale material handling system- belt conveying system.

5.1 Background

Belt conveyor system (BCS) is widely accepted as a major equipment in continues material handling domain. Its usages are well developed in various logistics domains, such as container/dry bulk terminals, airport and mine industry. Normally, the BCS is deployed in an open and harsh environment, as such, major components could suffer severe damage as system ages. Consequently, a monitoring and supervision system with decision support is essential to help users (from operations, maintenance, reliability and other departments) gain a consistent understanding about the system status and enabling effective planning and execution of maintenance. Due to the limit space of the paper, we demonstrate a typical fault supervision process- belt tear condition supervision which is made up for 85% among all system component damages for a BCS [34].

5.2 Belt Tear Condition Supervision

This scenario demonstrates the CASS capability of the system. Specifically, when inspection of tear shape is available, the system should analyse the damage level and propagation pace of the damage by intelligent supervision method, and create decisions in the form of possible maintenance activities and/or warning/alarm message if needed. We identify three key processes of implementing the context-aware supervision service for belt tear condition:

- Context modelling: It concerns extending the upper ontology (ontoSupervision) with definition of new entities for application purpose. Such extension is termed domain-specific ontology and partial illustration for BCS supervision (*ontoBeltCon*) is depicted in Fig. 6. The semantic meaning is given as: a tear shape (TS) *isMeasuredBy* a human inspection tool(HIT), which *isParameterOf* belt (belt Sect. 01). To supervise the tear condition, a belt tear supervision agent (BTSA) is designed. The BTSA *hasAssociatedMethod* belt tear supervision method (BTSM1). For supervision purpose, TS and a belt tear condition log (BTCL) *isSubscribedBy* BTSA. Upon successful decision making, a belt tear condition (BTC) is supervised which *isDeterminedBy* BTSM1 and *activates* alarm (alarm level 1). Finally the BTC *isResponisbleFor* user (maintenance operator 01). Besides newly added entities and its individuals, the data properties for a belt tear shape and belt tear condition is also available in Fig. 6.
- Context supervision: After context information are collected and pre-processed by ontology, the agent intelligence should be invoked. For a belt tear condition

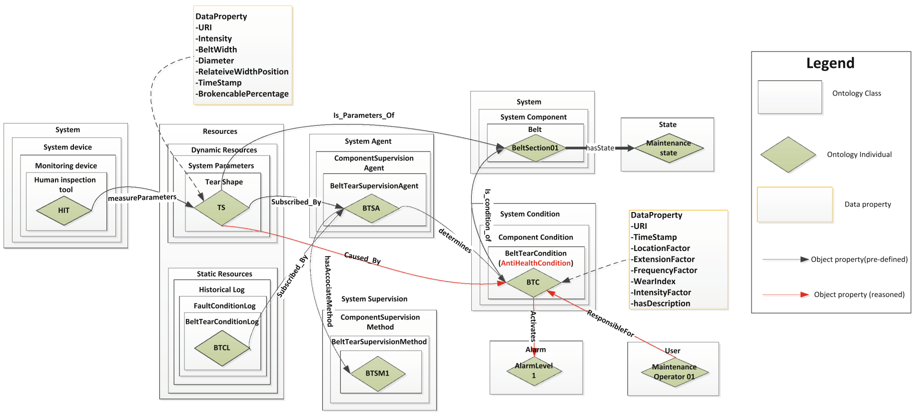

Fig. 6. *OntoBeltCon* configuration for belt tear condition supervision: include class, subclasses, data properties and object properties

supervision, a fuzzy logic based approach is applied [34]. Decisions are made based on the current tear shape measurement and history inspection log for the same shape. Two indicators (belt wear index and inspection frequency index) are provided to deliver a consistent understanding and interpretation of the supervision result and give straight forward suggestions for possible maintenance actions.

- Response actions: The supervised condition will be send back to ontology model for further processing before finally delivered to end users. In essence, it will use the supervision indicators to quantify the alarm level by running defined rules. For the given scenario, the rules can be given as Eqs. (1) and (2):

$$BeltTearCondition(?condition), greaterThan(?level, 0), hasWearIndex(?condition\\ , ?level), lessThanOrEqual(?level, 0.7) - > AntiHealthCondition(?condition) \tag{1}$$

$$BeltTearCondition(?condition), greaterThan(?level, 0), hasWearIndex(?condition\\ , ?level), greaterThanOrEqual(?level, 0.7) - > FaultCondition(?condition) \tag{2}$$

The semantic meaning is explained as: if a belt tear condition is confirmed and its wear index is go beyond 0.7, then a fault condition is inferred, while if the value is below 0.7 then an anti-health condition is given. Based on the triggered condition type, different alarm level is reasoned followed by rules given in Eq. (3):

$$\begin{aligned} &AntiHealthCondition(?condition) - > activates(?condition, AlarmLevel2)\\ &FaultCondition(?condition) - > activates(?condition, AlarmLevel3) \end{aligned} \tag{3}$$

5.3 Towards Implementation

In this section, several implementation issues are addressed. The overall implementation frameworks is depicted in Fig. 7.

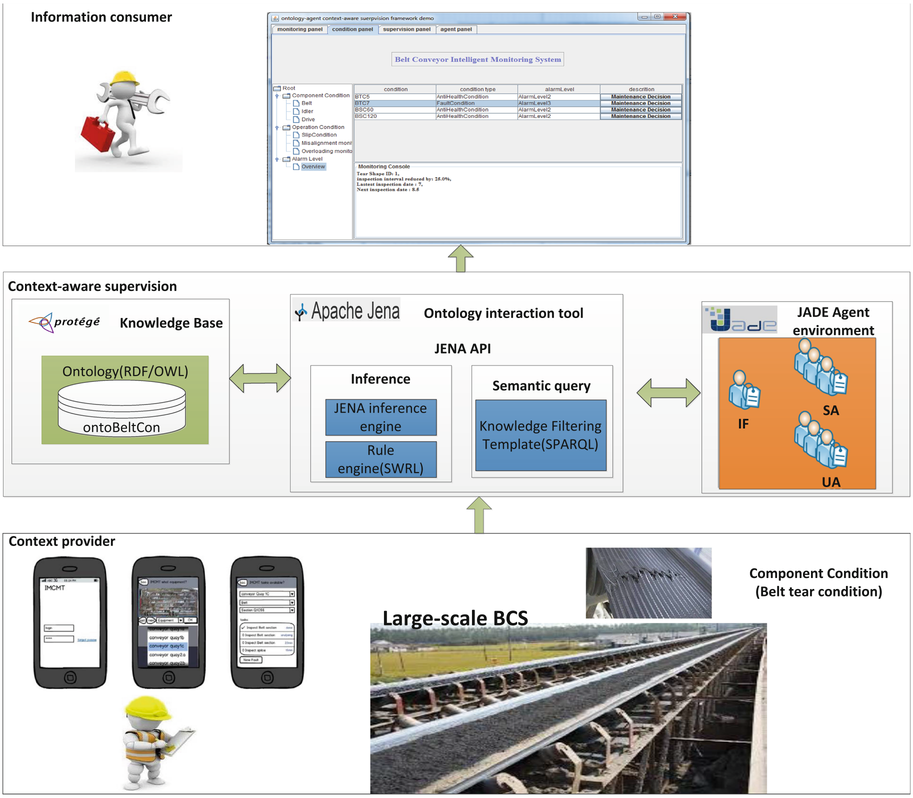

Fig. 7. Implementation architecture: ground layer: mobile inspection of tear shape [35]; middle layer: CASS; upper layer: GUI programmed uses Java-Swing

For the belt tear shape inspection, a mobile based tool is implemented [35]. It is used to assist human inspectors to register and record inspection data. The information will be pre-processed and send to CASS remotely for further processing.

The context model *ontoBeltCon* is created using open source tool Protege[1] and the ontology inference is realized via Pallet reasoner[2]. It has been saved as an owl file and persisted in remote server.

[1] Protégé, http://protege.stanford.edu/.

[2] Pallet, https://www.w3.org/2001/sw/wiki/Pellet.

For agent system implementation, a JAVA-based agent development environment (JADE)[3] tool is adapted. The JADE compliant with FIPA-standard[4] and provides an rich libraries and tools for controlling and managing agent behaviour. The supervision method is programmed in java and packaged in the agent behaviour. The inter-communication between agent is supported by the build-in ACL messages.

For agent-ontology integration, Apache-JENA[5] is used. It is a middle-ware that allow any Java-based program to interact with ontology. Through Jena, the JADE agent could interact with ontology and perform actions include create, update, read and delete (CRUD). And all the operations is done by implementing and executing SPARQL queries.

Finally, for demonstration purpose, a simple GUI is programmed by using Java swing to show how the system behaves.

6 Conclusion and Future Works

In this paper, a novel concept of context-aware supervision and its associated implementation techniques are proposed. The motivation behind the concept is to enable an efficient and transparent information flow for asset supervision tasks. We implement such system for supervision of a large-scale material handling system to demonstrate its major functionalities and potential usage in the domain of logistics. The designed system properly addresses the challenges mentioned in Sect. 1 as follows:

- Heterogeneity: A key concept of managing heterogeneity is to make such resources transparent. The use of ontology captures the capabilities and structure of such resource into a suitable representations and provide uniform storage platform that assist the work flows of heterogeneity environment.
- Interoperability: The data level interoperability is dealt with by means of ontology. The system level interoperability is addressed by establishing a uniform agent-ontology integration mechanism. The one in organization level is approached by means of context-aware service.
- Integrated decision making: The integrated decision making for asset supervision is achieved by designing and deploying a multi agent system that running and collaborating different intelligence together.

By integrating ontology with agent system, a novel ICT framework is created. The developed system together with the technological enablers are generic. As such we conclude that the system and concept is free to use in other similar domains. Future works include further extending the ontology model to incorporate more generic entities and concept in the system supervision domain. Moreover, to cope with more complex diagnosis/prognosis problem and enable more sophisticated decision making engine, the agent intelligence can be future investigated.

---

[3] JADE, https://jena.apache.org/.

[4] FIPA, http://www.fipa.org/.

[5] JENA, https://jena.apache.org/.

Acknowledgement. This research is supported by the China Scholarship Council under Grant 201307720072.

References

1. Neng Chiu, H.: The integrated logistics management system: a framework and case study. Int. J. Phys. Distrib. Logistics Manage. **25**(6), 4–22 (1995)
2. Gunasekaran, A., Ngai, E.W., Cheng, T.E.: Developing an e-logistics system: a case study. Int. J. Logistics **10**(4), 333–349 (2007)
3. De La Cruz, A.L., Veeke, H.P.M., Lodewijks, G.: Prognostics in the control of logistics systems. In: Proceedings of IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 1–5 (2006)
4. Takata, S., Kirnura, F., Van Houten, F.J.A.M., Westkamper, E., Shpitalni, M., Ceglarek, D., Lee, J.: Maintenance: changing role in life cycle management. CIRP Ann. Manuf. Technol. **53**(2), 643–655 (2004)
5. Thomas, G., Thompson, G.R., Chung, C.W., Barkmeyer, E., Carter, F., Templeton, M., Fox, S., Hartman, B.: Heterogeneous distributed database systems for production use. ACM Comput. Surv. (CSUR) **22**(3), 237–266 (1990)
6. Winters, L.S., Gorman, M.M., Tolk, A.: Next generation data interoperability: it's all about the metadata. In: IEEE Fall Simulation Interoperability Workshop (2006)
7. Clark, T., Jones, R.: Organisational interoperability maturity model for C2. In: Proceedings of the 1999 Command and Control Research and Technology Symposium (1999)
8. Chioreanu, A., Brad, S., Porumb, C., Porumb, S.: E-maintenance ontology-based approach for heterogeneous distributed robotic production capabilities. Int. J. Comput. Integr. Manuf. **28**(2), 200–212 (2015)
9. Candell, O., Karim, R., Söderholm, P.: eMaintenance—Information logistics for maintenance support. Rob. Comput. Integr. Manuf. **25**(6), 937–944 (2009)
10. Arnaiz, A., Iung, B., Adgar, A., Naks, T., Tohver, A., Tommingas, T., Levrat, E.: Information and communication technologies within E-maintenance. In: E-maintenance, pp. 39–60. Springer, London (2010)
11. Pistofidis, P., Emmanouilidis, C., Papadopoulos, A., Botsaris, P.N.: Modeling the semantics of failure context as a means to offer context-adaptive maintenance support. In: Proceedings of Second European Conference of the Prognostics and Health Management Society, pp. 8–10. PHME (2014)
12. Hong, J.Y., Suh, E.H., Kim, S.J.: Context-aware systems: a literature review and classification. Expert Syst. Appl. **36**(4), 8509–8522 (2009)
13. Galar, D., Thaduri, A., Catelani, M., Ciani, L.: Context awareness for maintenance decision making: a diagnosis and prognosis approach. Measurement **67**, 137–150 (2015)
14. Evchina, Y., Puttonen, J., Dvoryanchikova, A., Lastra, J.L.M.: Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. Eng. Appl. Artif. Intell. **43**, 111–126 (2015)
15. Pistofidis, P., Emmanouilidis, C.: Profiling context awareness in mobile and cloud based engineering asset management. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds.) APMS 2012. IAICT, vol. 398, pp. 17–24. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40361-3_3
16. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001)
17. Forkan, A., Khalil, I., Tari, Z.: CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living. Future Gener. Comput. Syst. **35**, 114–127 (2014)

18. Garrido, P.C., Ruiz, I.L., Gómez-Nieto, M.Á.: OBCAS: an agent-based system and ontology for mobile context aware interactions. J. Intell. Inform. Syst. **43**(1), 33–57 (2014)
19. El Kadiri, S., Grabot, B., Thoben, K.D., Hribernik, K., Emmanouilidis, C., von Cieminski, G., Kiritsis, D.: Current trends on ICT technologies for enterprise information systems. Comput. Ind. **79**, 14–33 (2016)
20. Kumar, U., Ahmadi, A., Verma, A.K., Varde, P.: Current Trends in Reliability, Availability, Maintainability and Safety: An industry Perspective. Springer, Switzerland (2015)
21. Hoareau, C., Satoh, I.: Modelling and processing information for context-aware computing: a survey. New Gener. Comput. **27**(3), 177–196 (2009)
22. Krummenacher, R., Strang, T.: Ontology-based context modelling. In: Proceedings (2007)
23. Schmohl, R., Baumgarten, U.: A generalized context-aware architecture in heterogeneous mobile computing environments. In: Proceedings of the Fourth International Conference on Wireless and Mobile Communications, ICWMC 2008, pp. 118–124 (2008)
24. Staab, S., Studer, R.: Handbook on Ontologies. Springer Science & Business Media (2013)
25. Kim, K., Kim, H., Kim, S.K., Jung, J.Y.: i-RM: An intelligent risk management framework for context-aware ubiquitous cold chain logistics. Expert Syst. Appl. **46**, 463–473 (2016)
26. Nadoveza, D., Kiritsis, D.: Ontology-based approach for context modelling in enterprise applications. Comput. Ind. **65**(9), 1218–1231 (2014)
27. Natarajan, S., Srinivasan, R.: Implementation of multi agents based system for process supervision in large-scale chemical plants. Comput. Chem. Eng. **60**, 182–196 (2014)
28. Feng, F., Pang, Y., Lodewijks, G.: Integrate multi-agent planning in hinterland transport: design, implementation and evaluation. Adv. Eng. Inform. **29**(4), 1055–1071 (2015)
29. Mahdavi, I., Shirazi, B., Ghorbani, N., Sahebjamnia, N.: IMAQCS: design and implementation of an intelligent multi-agent system for monitoring and controlling quality of cement production processes. Comput. Ind. **64**(3), 290–298 (2013)
30. Dawson, R.J., Peppe, R., Wang, M.: An agent-based model for risk-based flood incident management. Nat. Hazards **59**(1), 167–189 (2011)
31. Yu, R., Iung, B., Panetto, H.: A multi-agents based E-maintenance system with case-based reasoning decision support. Eng. Appl. Artif. Intell. **16**(4), 321–333 (2003)
32. Natarajan, S., Ghosh, K., Srinivasan, R.: An ontology for distributed process supervision of large-scale chemical plants. Comput. Chem. Eng. **46**, 124–140 (2012)
33. Dibley, M., Li, H., Rezgui, Y., Miles, J.: An ontology framework for intelligent sensor-based building monitoring. Autom. Constr. **28**, 1–14 (2012)
34. Lodewijks, G., Ottjes, J.A.: Application of fuzzy logic in belt conveyor monitoring and control. BeltCon **13**, 2–3 (2005)
35. Pang, Y., Lodewijks, G.: A remote intelligent belt conveyor inspection tool. In: Proceedings of 11th International Congress Bulk Materials Storage, Handling and Transportation. Newcastle, Australia (2013)
36. Feng, F., Pang, Y., Lodewijks, G.: An intelligent context-aware system for logistics asset supervision service. In: Proceedings of 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1147–1152. Gdansk, Poland (2016)